

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-202236  
(43)Date of publication of application : 27.07.2001

(51)Int.Cl. G06F 9/06  
H03K 19/173

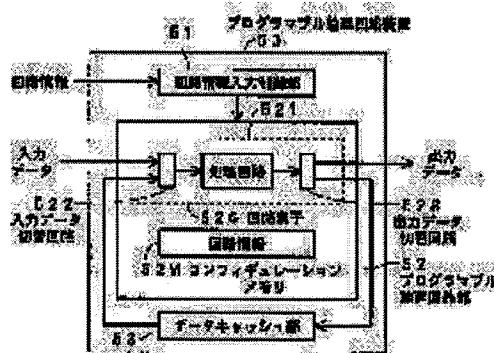
(21)Application number : 2000-014169 (71)Applicant : FUJI XEROX CO LTD  
(22)Date of filing : 20.01.2000 (72)Inventor : NISHIHARA YOSHIO  
SATO YOSHIHIDE  
YAMADA KIICHI  
SATONAGA  
TETSUKAZU

## (54) DATA PROCESSING METHOD FOR PROGRAMMABLE LOGIC CIRCUIT DEVICE AND THE SAME DEVICE AND INFORMATION PROCESSING SYSTEM AND CIRCUIT RECONSTITUTING METHOD FOR THE SAME DEVICE

### (57)Abstract:

PROBLEM TO BE SOLVED: To provide a cache logic technique for shortening an integral data processing time.

SOLUTION: In this programmable logic circuit device equipped with a circuit information input controlling part 51, a programmable logic circuit part 52, and a data cache part 53, blocked input data are successively processed by block units by plural circuits. The plural circuits are successively reconstituted in the programmable logic circuit device for each of the plural blocks which can be preserved in the data cache part 53. The intermediate data of the plural block units are preserved in the data cache part 53 so that the input data of the reconstituted circuits can be obtained, and the intermediate data as the processed results of the



reconstituted circuits are preserved so as to be overlapped in the data cache part 53. When the processing of the plural circuits is ended, the processed results are outputted to the outside part without being preserved in the data cache part 53.

---

## LEGAL STATUS

[Date of request for examination] 09.02.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-202236

(P2001-202236A)

(43)公開日 平成13年7月27日 (2001.7.27)

(51)Int.Cl.<sup>7</sup>

G 0 6 F 9/06  
H 0 3 K 19/173

識別記号

5 4 0

F I

G 0 6 F 9/06  
H 0 3 K 19/173

テ-マコ-ト(参考)

5 4 0 M 5 B 0 7 6  
5 J 0 4 2

審査請求 未請求 請求項の数12 O L (全 25 頁)

(21)出願番号

特願2000-14169(P2000-14169)

(22)出願日

平成12年1月20日 (2000.1.20)

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72)発明者 西原 義雄

神奈川県足柄上郡中井町境430 グリーン  
テクなかい富士ゼロックス株式会社内

(72)発明者 佐藤 嘉秀

神奈川県足柄上郡中井町境430 グリーン  
テクなかい富士ゼロックス株式会社内

(74)代理人 100091546

弁理士 佐藤 正美

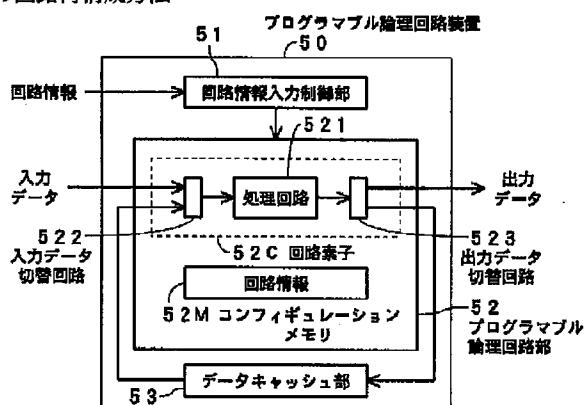
最終頁に続く

(54)【発明の名称】 プログラマブル論理回路装置によるデータ処理方法、プログラマブル論理回路装置、情報処理システム、プログラマブル論理回路装置への回路再構成方法

(57)【要約】

【課題】 総合のデータ処理時間を短縮することができるキャッシュロジック技術を実現する方法を提供する。

【解決手段】 回路情報入力制御部51と、プログラマブル論理回路部52と、データキャッシュ部53とを備えるプログラマブル論理回路装置を用いて、ブロック化されている入力データを、複数個の回路でそれぞれブロック単位で処理を順次に行う。データキャッシュ部53に保存可能な複数個のブロック毎に、複数個の回路を順次にプログラマブル論理回路装置に再構成して処理を行う。複数個のブロックの単位の中間データは、データキャッシュ部53に保存して、再構成された回路の入力データとし、その再構成された回路の処理結果としての中間データは、データキャッシュ部53に上書き保存する。複数個の回路の処理が終了したら、その処理結果は、データキャッシュ部53に保存することなく、外部に出力する。



## 【特許請求の範囲】

【請求項1】回路素子と、この回路素子に接続されるコンフィギュレーションメモリとを備え、コンフィギュレーションメモリに書き込まれる回路情報に基づいて前記回路素子が用いられて回路が構成されるプログラマブル論理回路部を備えると共に、前記回路情報に基づいて構成された回路で処理されるデータを一時的に保持するデータキャッシュ手段を備えるプログラマブル論理回路装置を用いて、ブロック化されている入力データを、複数個の回路でそれぞれ前記ブロック単位で処理を順次に行うデータ処理方法において、

前記複数個の回路のうちの最初の回路の回路情報を前記コンフィギュレーションメモリに書き込み、前記プログラマブル論理回路部に前記最初の回路を構成する第1コンフィギュレーション工程と、

前記データキャッシュ手段で保存可能な前記入力データの複数ブロックを、前記最初の回路で処理し、その処理結果のブロック単位のデータを前記データキャッシュ手段に保存する第1処理工程と、

前記データキャッシュ手段に、前記入力データの複数ブロック分の処理結果が保存された後に、前記複数個の回路のうちの次の回路の回路情報を前記コンフィギュレーションメモリに書き込み、前記プログラマブル論理回路部に前記次の回路を構成する次コンフィギュレーション工程と、

前記データキャッシュ手段に保存された複数ブロック分の処理結果のデータを、前記次コンフィギュレーション工程で構成された回路で処理し、その処理結果の複数ブロック分のデータを前記データキャッシュ手段に保存する次処理工程と、

前記次コンフィギュレーション工程と、前記次処理工程とを前記複数個の回路の最後の回路まで繰り返し、前記最後の回路の処理結果を出力データとして外部に出力する出力処理工程と、

を備え、

前記第1コンフィギュレーション工程から前記出力処理工程までを、前記入力データの複数ブロック分ごとに、前記入力データの全てについて行うことを特徴とするプログラマブル論理回路装置によるデータ処理方法。

## 【請求項2】請求項1において、

処理された入力データのブロック数を記憶し、前記入力データの全てのデータについての処理の終了を検知するためのデータカウント工程を備えることを特徴とするプログラマブル論理回路装置によるデータ処理方法。

## 【請求項3】請求項1において、

前記データキャッシュ手段に格納されたブロック数を記憶し、処理サイクルの終了を検知するためのキャッシュカウント工程を備えることを特徴とするプログラマブル論理回路装置によるデータ処理方法。

## 【請求項4】アプリケーションプログラムの処理の少な

くとも一部分を、プログラマブル論理回路装置で処理する情報処理システムであって、

プログラマブル論理回路装置に対して、請求項1～請求項3に記載されたデータ処理方法のいずれかを用いることを特徴とする情報処理システム。

【請求項5】回路素子と、この回路素子に接続されるコンフィギュレーションメモリとを備え、前記コンフィギュレーションメモリに書き込まれる回路情報に基づいて回路が構成されるプログラマブル論理回路部と、前記コンフィギュレーションメモリに回路情報を書き込むための回路情報入力制御手段と、

前記コンフィギュレーションメモリに書き込まれた回路情報に基づいて構成された回路で処理されるデータを一時的に保持するデータキャッシュ手段と、

を備えることを特徴とするプログラマブル論理回路装置。

【請求項6】回路素子と、この回路素子に接続されるコンフィギュレーションメモリとを備え、前記コンフィギュレーションメモリに書き込まれる回路情報に基づいて回路が構成されるプログラマブル論理回路部と、

前記コンフィギュレーションメモリに回路情報を書き込むための回路情報入力制御手段と、

前記コンフィギュレーションメモリに書き込まれた回路情報に基づいて構成された回路で処理されるデータを一時的に保持するデータキャッシュ手段と、

前記コンフィギュレーションメモリに書き込まれた回路情報に基づいて構成される回路に対して、前記データキャッシュ手段からのデータと、入力データとのいずれか一方を供給するようにするための入力データ切替手段と、

前記コンフィギュレーションメモリに書き込まれた回路情報に基づいて構成される回路の出力データを、前記データキャッシュ手段に供給するか、あるいは出力データとして導出するかの切り替えを行うための出力データ切替手段と、

を備えることを特徴とするプログラマブル論理回路装置。

## 【請求項7】請求項5または請求項6において、

前記データキャッシュ手段は、

前記コンフィギュレーションメモリに書き込まれた第1の回路情報に基づいて構成される回路からの出力データを保持し、この保持しているデータを、前記コンフィギュレーションメモリに書き込まれた第2の回路情報に基づいて構成される回路に、その入力データとして出力することを特徴とするプログラマブル論理回路装置。

## 【請求項8】請求項5または請求項6において、

前記データキャッシュ手段は、前記プログラマブル論理回路部の前記回路素子を用いて前記プログラマブル論理回路部内に設けることを特徴とするプログラマブル論理回路装置。

**【請求項 9】**請求項 6において、

前記入力データ切替手段および前記出力データ切替手段は、前記プログラマブル論理回路部の前記回路素子を用いて前記プログラマブル論理回路部内に設けることを特徴とするプログラマブル論理回路装置。

**【請求項 10】**請求項 5 または請求項 6において、前記回路情報入力手段は、入力される回路情報を前記コンフィギュレーションメモリの少なくとも一部に書き込むことにより、前記プログラマブル論理回路部の前記回路素子の少なくとも一部の回路構成を変更することを特徴とするプログラマブル論理回路装置。

**【請求項 11】**アプリケーションプログラムの処理の少なくとも一部分を、プログラマブル論理回路装置で処理する情報処理システムであって、前記プログラマブル論理回路装置に、請求項 5～請求項 10 のいずれかに記載のプログラマブル論理回路装置を用いたことを特徴とする情報処理システム。

**【請求項 12】**回路素子と、この回路素子に接続されるコンフィギュレーションメモリとを備え、コンフィギュレーションメモリに書き込まれる回路情報に基づいて前記回路素子が用いられて回路が構成されるプログラマブル論理回路部を備えると共に、前記回路情報に基づいて構成された回路で処理されるデータを一時的に保持するデータキャッシュ手段を備えるプログラマブル論理回路装置への回路再構成方法であって、

ブロック化されている入力データを、複数個の回路で順次に処理して出力するもので、各回路ではそれぞれ前記ブロック単位で処理を行うようにする際に、

前記複数個の回路うちの最初の回路の回路情報に基づいて前記最初の回路を、前記プログラマブル論理回路部に構成する最初回路構成工程と、

前記プログラマブル論理回路部に構成した回路が、前記データキャッシュ手段で保存可能な前記入力データの複数ブロックを処理し、その処理結果をデータキャッシュ手段に保持した後に、次の回路の回路情報に基づいて前記次の回路を構成する次回路構成工程と、

前記次回路構成工程を、前記複数個の回路のうちの最後の回路まで繰り返す工程と、

前記プログラマブル論理回路部に前記最後の回路が構成され、前記最後の回路の処理結果が出力データとして外部に出力された後、前記入力データの終了まで、前記最初回路工程以下を繰り返すようにすることを特徴とするプログラマブル論理回路装置への回路再構成方法。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】**この発明は、例えばアプリケーションプログラムによる処理の一部分を、回路構成を再構成できるプログラマブル論理回路装置を用いてハードウェア処理することが可能である情報処理システムおよびこの情報処理システムでのデータ処理方法並びにこ

の情報処理システムに用いる前記プログラマブル論理回路装置に関する。特に、プログラマブル論理回路装置での再構成時間も含めた処理時間を短縮する方法に関する。

**【0002】**

**【従来の技術】**デジタル回路装置の分野において、フィールドプログラマブルゲートアレイ (FPGA) やプログラマブルロジックデバイス (PLD) などのプログラマブル論理回路装置が、特定用途向け集積回路 (ASIC) を作成する前の試作デバイスとして、または、数週間から数ヶ月という長い作製期間を要するASICの代替デバイスとして使われてきた。また、最近では、回路装置作成後に仕様を変更したり、回路を修正することを可能にするために、プログラマブル論理回路装置が使われている。

**【0003】**一般的なプログラマブル論理回路装置の構造を図17に示す。プログラマブル論理回路装置1は、回路情報を外部から読み込む回路情報入力制御部2と、読み込まれた回路情報に従って回路機能を実現するプログラマブル論理回路部3とから構成される。

**【0004】**さらに、プログラマブル論理回路部3の詳細な構造は、図18に示すように、回路素子4と、回路素子4に接続するコンフィギュレーションメモリ5とから構成される。回路素子4は、入出力部、論理回路セル、配線部からなるが、この回路素子4の接続形態により、プログラマブル論理回路装置は、FPGA型と CPLD型とに分けられる。

**【0005】**FPGA型のプログラマブル論理回路部3Aは、図19(A)に示すように、2次元格子状に配列された論理回路セル6Aが、配線部7Aにより相互に接続される。また、全体として矩形の配線部7Aの四辺に接続された入出力部8Aを介して外部と信号の入出を行なう。

**【0006】**また、CPLD型のプログラマブル論理回路部3Bは、図19(B)に示すように、入出力部8Bと論理回路セル6Bが、ツリー構造状に配線部7Bに接続される。

**【0007】**いずれの構造においても、プログラマブル論理回路装置1に読み込まれた回路情報は、回路情報入力制御部2によってコンフィギュレーションメモリ5に書き込まれる。コンフィギュレーションメモリ5に書き込まれた回路情報に従って、回路素子の機能と結線状態が決められる。この操作をプログラマブル論理回路装置の再構成またはコンフィギュレーションと呼ぶ。

**【0008】**従来のプログラマブル論理回路装置では、回路情報を読み込むたびにコンフィギュレーションメモリ全体が書き換えられて、プログラマブル論理回路部に構成された回路全体が再構成される。

**【0009】**最近では、コンフィギュレーションメモリの一部分に対応する回路情報のみを読み込むことが可能

になってきた。その結果、プログラマブル論理回路装置で動作中の回路の一部分を変更したり、動作中の回路を停止することなく、プログラマブル論理回路部に新たな回路を追加することができるようになった。このとき、プログラマブル論理回路装置上にある処理の中間データが失われることもない。このようなプログラマブル論理回路装置は、動的に部分再構成可能なプログラマブル論理回路装置と呼ばれる。

【0010】【プログラマブル論理回路装置の新しい応用】インターネットに代表されるデジタル通信ネットワークの発展と普及に伴い、デジタル通信方式と、これを用いてネットワーク上で流通されるデジタルメディア方式の開発と標準化が急速に進んでいる。これらの方に従ってネットワーク上のデジタル信号を処理する装置は、処理を行うデバイスの視点から大きく二つに分類することができる。

【0011】ひとつは、プログラムで記述された手順に従って汎用プロセッサを用いて処理するソフトウェア処理であり、もうひとつは、ASICなどの専用処理回路を用いて回路結線で記述された手順に従って処理するハードウェア処理である。

【0012】ソフトウェア処理は、プログラムを変更することにより、ひとつのプロセッサが複数のデータを処理したり、新たな方式に対応することができるという特長をもつ。その反面、プログラムが格納されているメモリから命令をフェッチしてデコードしたり、実行結果をメモリにライトバックするというオーバーヘッドがあるので、同じクロック周波数で動作するハードウェア処理に比べて処理速度が遅い。また、プログラムを格納する主記憶メモリや二次記憶装置が必要となるので、処理装置が大きくなるという欠点がある。

【0013】一方、ハードウェア処理は、処理手順の記述が回路結線で実現されているので、同じクロック周波数で動作するソフトウェア処理に比べて処理のオーバーヘッドが少ないため、処理速度が早く、また、処理手順を記憶するメモリが不要なので処理装置が小さい、という特長をもつ。その反面、一度作製した回路結線は変更することができないので、ソフトウェア処理に比べて柔軟性に欠け、複数の方式のデータを処理するためには複数の専用処理回路が必要であり、一度作製した回路は新たな方式には対応できない。

【0014】ハードウェア処理が柔軟性に欠けるという欠点を解決するために、プログラマブル論理回路装置を用いたハードウェア処理が最近注目されている。すなわち、処理速度が早く、処理装置が小さいというハードウェア処理の特長を保ちつつ、プログラマブル論理回路装置の回路情報を適宜変更することにより、複数の方式や新しい方式に対応しようとするものである。

【0015】このように、プログラマブル論理回路装置によるハードウェア処理により、汎用プロセッサを用い

たソフトウェア処理と同じように柔軟性を持ち、ソフトウェア処理よりも高速な処理を実現する技術は、リコンフィギュラブルコンピューティングと呼ばれている。

【0016】【リコンフィギュラブルコンピューティング技術の説明】リコンフィギュラブルコンピューティングでは、アプリケーション処理で必要となる複数の処理回路の回路情報を外部記憶装置へ事前に格納しておき、必要に応じて、その外部記憶装置から読み出した回路情報を、プログラマブル論理回路装置内のコンフィギュレーションメモリに書き込むことで、必要とする回路をプログラマブル論理回路装置上に実現する。

【0017】この技術は、必要な回路情報をプログラマブル論理回路装置の外部に退避しておくという観点から、キャッシュロジック技術と呼ばれたり、回路情報を書き替えることにより、実際のプログラマブル論理回路部の回路規模よりも大きな規模の回路を実現できるという観点から、バーチャルロジック技術と呼ばれている。以下の説明では、簡単のため、これらの技術を総称して、キャッシュロジック技術と呼ぶことにする。

【0018】キャッシュロジック技術は、同じプログラマブル論理回路装置上に、必要に応じて異なる回路を構成するという時分割駆動技術である。その結果、回路規模の小さなプログラマブル論理回路装置を用いて、その回路規模以上の回路を実現でき、回路装置の小型化と低コスト化が可能となる。

【0019】リコンフィグラブルコンピューティング技術の例として、特開平10-78932号公報に記載される「リコンフィグラブル・ネットワークコンピュータ」があり、それを従来例1として図20を参照しながら説明する。

【0020】この従来例1の情報処理システムは、通信ネットワークNETに接続された複数のコンピュータで構成され、その内の少なくとも一つがアプリケーションプログラムを配布するコンピュータ（アプリケーション・サーバ）SBであり、残りのコンピュータが前記アプリケーションプログラムをダウンロードし、実行するクライアントとなるコンピュータ（クライアント・コンピュータ）CLで構成されたシステムである。複数のクライアント・コンピュータCLの一部に、プログラムにより機能を随时変更し再構成することが可能な拡張ハードウェア（拡張HW）11が搭載されている。

【0021】アプリケーション・サーバSBに格納されたアプリケーションプログラムAPには、その一部の機能に対して、拡張ハードウェアで実行されるプログラムコード（拡張コード）と、クライアント・コンピュータCLのメインプロセッサ（メインP）12で実行されるメインプロセッサコードが含まれている。

【0022】クライアント・コンピュータCLのOSは、拡張ハードウェア11が実装されているか否かを判断する機能を持ち、その判断に基づいて、ハードウェア

構成に適したコードをアプリケーションプログラムAPから取り出すコード選択機能13を備える。そして、図20の上側のクライアント・コンピュータCLのように、拡張ハードウェア11が実装されている場合には、コード選択機能13を通じてアプリケーションプログラムAPの中から拡張コードが取り出されて、拡張ハードウェアで処理が実行される。

【0023】また、図20の下側のクライアント・コンピュータCLのように、拡張ハードウェア11を持たない場合には、アプリケーションプログラムAPから、コード選択機能13を通じてメインプロセッサコードが取り出されて、メインプロセッサ12で処理が実行される。

【0024】この従来例1によれば、ネットワークで接続されたコンピュータ上で、サーバから配布されたアプリケーションプログラムをクライアント側で実行する際、クライアント側に、プログラムにより機能を随時変更し再構成可能な拡張ハードウェアを搭載し、サーバに格納されたアプリケーションプログラムには、クライアントのメインプロセッサコードと拡張コードを含ませ、拡張ハードウェアの有無、種類を判断したコード選択機能によって、クライアント側の計算機の構成を変え、処理に適した構成にすることでアプリケーションプログラムを高速に処理できる。

【0025】しかしながら、この従来例1の場合、プログラマブル論理回路装置のコンフィギュレーションメモリに書き込む回路情報（拡張コード）の規模によっては、アプリケーションサーバSBからクライアントコンピュータCLのプログラマブル論理回路装置のコンフィギュレーションメモリに回路情報を書き込む時間が長くなり、専用のハードウェア処理回路である拡張ハードウェアを用いて高速処理を実現しても、回路再構成時間を含めた全体の処理時間が、ソフトウェア処理時間よりも長くなるという問題がある。

【0026】この問題のひとつの解決方法が、マルチコンテキスト技術と呼ばれるデバイス技術である。すなわち、マルチコンテキスト技術においては、プログラマブル論理回路装置内に複数の回路情報を格納することができるよう、複数個のコンフィギュレーションメモリを設け、必要に応じてコンフィギュレーションメモリを切り替えることにより、プログラマブル論理回路装置に回路を再構成することで、回路の再構成時間を大幅に短縮する。

【0027】【マルチコンテキスト技術によるプログラマブル論理回路装置の説明】マルチコンテキスト技術によるプログラマブル論理回路装置の構造を、図21に示す。マルチコンテキスト技術によるプログラマブル論理回路装置20は、複数の回路情報を外部から読み込む回路情報入力制御部21と、複数の回路情報から必要な回路情報を選択する回路情報選択制御部22と、選択され

た回路情報に従って回路機能を実現するプログラマブル論理回路部23から構成される。

【0028】このマルチコンテキスト技術によるプログラマブル論理回路部23の詳細な構造は図22に示すようなものとなり、プログラマブル論理回路部23は、前述の場合と同様に、入出力部、論理回路セル、配線部からなる回路素子24と、この回路素子24に接続するコンフィギュレーションメモリ25とから構成されるが、このマルチコンテキスト技術によるプログラマブル論理回路部23の場合のコンフィギュレーションメモリ25は、複数個のメモリプレーンで構成される。

【0029】そして、このマルチコンテキスト技術によるプログラマブル論理回路部23の場合には、FPGA型とCPLD型（図19参照）とのいずれの構造においても、プログラマブル論理回路装置20に外部から読み込まれた複数の回路情報は、回路情報入力制御部21によってコンフィギュレーションメモリ25の各メモリプレーンに一つの回路情報が書き込まれる状態で書き込まれる。

【0030】そして、コンフィギュレーションメモリ25の複数個のメモリプレーンに書き込まれた複数の回路情報のうち、回路情報選択制御部22からの選択信号により選択されたメモリプレーンの回路情報に従って、回路素子24の機能と結線状態が決められて、プログラマブル論理回路装置20に回路が再構成される。

【0031】マルチコンテキスト技術の例として、FCCM'97 (FPGAs for Custom Computing Machines 1997) で発表された“A Time-Multiplexed FPGA”的例がある。これを従来例2として、図23を参照しながら説明する。

【0032】図23は、発表された時分割駆動FPGAの構成を示すものである。この時分割駆動FPGAは、米国ザイリンクス（Xilinx）社のXC4000Eを改良したものであり、回路素子31の論理セルと内部配線とを決定する、SRAMで構成されたコンフィギュレーションメモリ32を8組備えている。コンフィギュレーションメモリ32のそれぞれには、異なる回路構成に対応する回路情報が格納されており、これらのコンフィギュレーションメモリ32を切り替えることにより、時分割でFPGAの回路構成を再構成することができる。

【0033】上述の従来例2で示したように、マルチコンテキスト技術は、事前に複数の回路情報をコンフィギュレーションメモリに格納してあるので、回路の再構成時間を短縮することを可能にする。

【0034】しかしながら、回路情報を格納するためには、複数プレーンまたは複数個のコンフィギュレーションメモリがプログラマブル論理回路装置の内部に必要なため、プログラマブル論理回路部の回路規模が大きくな

る。回路規模が大きくなると、回路素子の負荷容量が大きくなることにより、回路性能が低下したり、消費電力が大きくなるという問題を引き起こす。また、回路規模が大きくなると、プログラマブル論理回路装置の製造コストが高くなるという問題も引き起こす。

#### 【0035】

【発明が解決しようとする課題】ところで、画像データなどを処理する情報処理システムでは、画像データを所定数の画素データの群からなるブロック単位で、複数個の処理回路について順次に処理するようにすることが多い。例えば、画像圧縮符号化処理を行う場合には、画像データをブロック化し、そのブロック化したデータについて、例えば直交変換回路で直交変換し、その直交変換後のデータについて量子化回路で量子化処理を行い、さらに可変長符号化（エントロピー符号化）回路で、可変長符号化処理を行う。

【0036】この場合に、一般的には、画像データは、順次に、1ブロック毎に、複数の処理回路に順次に供給し、各処理回路では、その1ブロック単位で処理を行い、1ブロック毎の出力信号を得る。そして、それらの処理を全ブロックについて繰り返し行うようとする。

【0037】このため、これらの複数個の回路処理を、キャッシュロジック技術やマルチコンテキスト技術を用いて、プログラマブル論理回路装置で実行使用とする場合には、上述したいずれの従来例のプログラマブル論理回路装置を用いる場合でも、ブロック単位に、直交変換回路、量子化回路、可変長符号化回路などの処理回路を、1ブロック単位のデータごとについて、順にプログラマブル論理回路部に再構成して、処理を行うようになるのが一般的である。

【0038】しかしながら、この方式では、プログラマブル論理回路部への回路の再構成回数が、処理対象のブロック数分必要になり、回路のコンフィギュレーション時間が、全体の処理時間に影響を与えて、全体の処理時間が長くなってしまって、回路再構成時間を含めた総処理時間で比べると、ソフトウェア処理よりも処理時間が遅くなるおそれがある。

【0039】前述した従来のプログラマブル論理回路装置を用いた上述の処理時間について、図を参照して、さらに詳しく説明する。

【0040】以下では、3つの処理回路C1、C2、およびC3からなるアプリケーションを例にして説明する。処理対象のデータは、Nブロック（Nは2以上の整数）のデータからなり、データを、処理回路C1、C2、C3で、順次、処理することで処理が完了する。

【0041】アプリケーションが、例えば画像のJPEG圧縮の場合であれば、処理回路C1、C2、C3が、それぞれDCT変換回路、量子化回路、エントロピー符号化回路に対応し、1ブロックのデータが、1画素が8ビット階調の $8 \times 8 = 64$ 個の画素データに対応する。

【0042】【従来のキャッシュロジック技術による回路の再構成】図17を用いて説明したように、この場合のプログラマブル論理回路装置1は、回路情報入力制御部2と、回路素子4およびコンフィギュレーションメモリ5を内蔵するプログラマブル論理回路部3とで構成される。

【0043】この例では、図24に示すように、回路C1、C2、C3のそれぞれを構成するための回路情報CD1、CD2、CD3が、回路情報入力制御部2を介してプログラマブル論理回路部3のコンフィギュレーションメモリ5に順次読み込まれて、回路素子4の部分に処理回路C1、C2、C3が構成され、この構成された処理回路C1、C2、C3のそれぞれで、Nブロックの入力データDin1、Din2、…、DinNを、順次処理して、Nブロックの出力データDout1、Dout2、…、DoutNを得る。

【0044】この処理手順を、図25に示したタイミングチャートと、図26に示したフローチャートを用いて詳しく説明する。

【0045】図26に示すように、処理が開始されると、図24に示していないアプリケーション制御部により、入力データのブロック番号を指示するデータカウンタの値が「1」にリセットされる（ステップS101）。

【0046】次に、回路情報CD1が、回路情報入力制御部2を介してコンフィギュレーションメモリ5に読み込まれて、プログラマブル論理回路部3に処理回路C1が構成される（ステップS102）。これが図25のタイミングチャートで最初の「読」として示した読み込み実行状態である。

【0047】こうして処理回路C1が構成されると、データカウンタが指示する入力データのブロックDin1が、処理回路C1に入力される（ステップS103）。データが入力されると、アプリケーション制御部によりデータカウンタの値に1が加えられる（ステップS104）。そして、入力データDin1が、処理回路C1で処理される（ステップS105）。データの入力から処理回路C1による処理までが、図25のタイミングチャートで最初の「処理」として示したデータ処理実行状態である。

【0048】ここで、回路情報CD1により構成される処理回路C1は、図27に示すように、入力データバッファ42と処理実行回路41とで構成される。入力データDin1は、一時的に入力データバッファ42に保持され、順次、処理実行回路41で処理される。処理された結果は、中間データとして、再び、入力データバッファ42に保持される。このとき、最初の入力データDin1は、中間データで上書きされて失われる。

【0049】以上のようにして、入力データDin1が処理回路C1で処理された中間データが、プログラマ

ル論理回路部3上の入力データバッファ42に保持された状態で、回路情報CD2が回路情報入力制御部2を介してコンフィギュレーションメモリ5に読み込まれて、処理回路C2がプログラマブル論理回路部3に構成される(ステップS106)。これが、図25のタイミングチャートにおいて、第2番目の「読み」で示す読み込み状態である。

【0050】処理回路C2が構成されると、プログラマブル論理回路部3上に保持されていた中間データが、処理回路C2に入力され、処理される(ステップS107)。中間データの入力から処理回路C2による処理までが、図25のタイミングチャートにおいて第2番目の「処理」として示したデータ処理実行状態である。

【0051】この場合に、図28に示すように、回路情報CD2は、入力データバッファ42を保持した状態で、処理回路C1の処理実行回路41を、処理実行回路43へと動的に部分再構成して、処理回路C2を構成する。その結果、中間データが、入力データバッファ42に保持され、処理回路C2の入力データとなる。そして、処理実行回路43で処理された結果は、新たな中間データとして再び入力データバッファ42に保持される。このとき、最初の中間データは、新たな中間データで上書きされて失われる。

【0052】処理回路C2で処理された結果が新たな中間データとしてプログラマブル論理回路部3上に保持された状態で、回路情報CD3が、回路情報入力制御部2を介してコンフィギュレーションメモリ5に読み込まれて、処理回路C3がプログラマブル論理回路部3に構成される(ステップS108)。これが、図25のタイミングチャートで、第3番目の「読み」として示した読み込み状態である。

【0053】処理回路C3が構成されると、プログラマブル論理回路部3上の入力データバッファ42に保持されていた中間データが、処理回路C3に入力され、処理され(ステップS109)、その処理結果が出力データDout1として出力される(ステップS110)。中間データの入力から処理回路C3による処理結果の出力までが、図25のタイミングチャートで、第3番目の「処理」として示したデータ処理実行状態である。

【0054】この場合、図29に示すように、回路情報CD3は、入力データバッファ42を保持した状態で、処理回路C2の処理実行回路43を処理実行回路44へと動的に部分再構成し、さらに出力データバッファ45を追加して、処理回路C3を構成する。その結果、中間データが入力データバッファ42に保持され、処理回路C3の入力データとなる。処理実行回路44で処理された結果は、出力データバッファ45に一時的に保持された後に、出力データとして出力される。

【0055】処理回路C3による処理結果が出力されたときに、データカウンタの値が、入力データの全てのブ

ロック数Nより小さい場合は、再び、回路情報CD1を読み込むことから初めて、すべての入力データを処理し終えるまで、上述のステップS102からステップS110までの処理サイクルを繰り返す(ステップS111)。

【0056】このように、3回の回路情報の読み込みと、それに引き続くデータ処理により、1ブロックのデータが処理される。このサイクルを、入力データのブロック数Nだけ繰り返すことにより、すべての入力データが処理される。

【0057】なお、図25と図26には示されていないが、回路情報の読み込み時や処理回路での処理時にエラーが発生した場合は、アプリケーション制御部にエラーの発生を通知して処理は終了する。

【0058】上述の例では、回路情報CD1が入力データバッファを生成し、回路情報CD3が出力データバッファを生成して、処理回路C1、C2、C3の中間データの保持を行うようにしたが、処理回路C1、C2、C3が中間データを保持する構成としては、これに限らない。処理回路C1、C2、C3が中間データを保持する別の回路構成の例を図30、図31、図32に示す。

【0059】この別の例においては、回路情報CD1により構成される処理回路C1は、図30に示すように、処理実行回路41、左データバッファ42L、右データバッファ42Rで構成される。入力データは、一時的に左データバッファ42Lに保持され、順次、処理実行回路41で処理される。処理された結果は、中間データとして右データバッファ42Rに保持される。このとき、最初の入力データは、左データバッファ42Lに保持されたままである。

【0060】回路情報CD2は、図31に示すように、左データバッファ42Lと右データバッファ42Rとを保持した状態で、処理回路C1の処理実行回路41を処理実行回路43へと動的に部分再構成して、処理回路C2を構成する。このとき、処理実行回路41とは異なり、処理実行回路43は、右データバッファ42Rを入力とし、左データバッファ42Lを出力とする。その結果、右データバッファ42Rに保持されていた中間データが、処理実行回路43の入力データとなる。処理実行回路43で処理された結果は、新たな中間データとして左データバッファ42Lに保持される。このとき、左データバッファ42Lに保持されていた最初の入力データは、新たな中間データで上書きされて失われる。

【0061】回路情報CD3は、図32に示すように、左データバッファ42Lと右データバッファ42Rとを保持した状態で、処理回路C2の処理実行回路43を処理実行回路44へと動的に部分再構成して、処理回路C3を構成する。このとき、処理実行回路44は、処理実行回路41と同じように、左データバッファ42Lを入力とし、右データバッファ42Rを出力とする。その結

果、左データバッファ42Lに保持されていた中間データが、処理実行回路44の入力データとなる。処理実行回路44で処理された結果は、右データバッファ42Rに一時的に保持された後に、出力データとして出力される。

【0062】【マルチコンテキスト技術を用いたリコンフィギュラブルコンピューティングにおける回路の再構成】図21を用いて説明したように、マルチコンテキスト技術によるプログラマブル論理回路装置は、複数の回路情報を外部から読みこむ回路情報入力制御部21と、複数の回路情報から必要な回路情報を選択する回路情報選択制御部22と、選択された回路情報を従って回路機能を実現するプログラマブル論理回路部23とから構成される。

【0063】マルチコンテキスト技術によるプログラマブル論理回路装置のアプリケーションの例では、図33に示すように、回路情報CD1、CD2、CD3が、回路情報入力制御部21を介してプログラマブル論理回路部23のコンフィギュレーションメモリ25に順次読み込まれて格納される。

【0064】そして、選択信号S1、S2、S3に従って回路情報選択制御部22により構成される処理回路C1、C2、C3で、Nブロックの入力データDin1、Din2、…、DinNを順次処理して、Nブロック分の出力データDout1、Dout2、…、DoutNを得る。

【0065】この処理手順を、図34に示したタイミングチャートと、図35に示したフローチャートを用いて詳しく説明する。

【0066】図35に示すように、処理が開始されると、図33に示していないアプリケーション制御部により、入力データのブロック番号を指し示すデータカウンタの値が「1」にリセットされる（ステップS201）。

【0067】次に、3つの回路情報CD1、CD2、CD3が、回路情報入力制御部21を介して順次読み込まれて、コンフィギュレーションメモリ25に順次格納される（ステップS202、203、204）。これが図34のタイミングチャートで、「読み込み」として示した読み込み実行状態である。

【0068】次に、選択信号S1に従って、回路情報選択制御部22の指示により処理回路C1が構成され（ステップS205）、データカウンタが指し示す入力データのブロックDin1が処理回路C1に入力される（ステップS206）。データが入力されると、アプリケーション制御部によりデータカウンタの値に「1」が加えられる（ステップS207）。そして、入力データDin1が処理回路C1で処理される（ステップS208）。

【0069】次に、入力データDin1が処理回路C1

で処理された中間データがプログラマブル論理回路部23上に保持された状態で、選択信号S2に従って回路情報選択制御部22の指示により、処理回路C2がプログラマブル論理回路部23に構成される（ステップS209）。この処理回路C2が構成されると、プログラマブル論理回路部23上に保持されていた中間データが処理回路C2に入力され、処理される（ステップS210）。

【0070】中間データが処理回路C2で処理された結果が、新たな中間データとしてプログラマブル論理回路部23上に保持された状態で、選択信号S3に従って回路情報選択制御部22の指示により、処理回路C3がプログラマブル論理回路部23に構成される（ステップS211）。この処理回路C3が構成されると、プログラマブル論理回路部23上に保持されていた中間データが、処理回路C3に入力され、処理される（ステップS212）。そして、その処理結果が、出力データDout1として出力される（ステップS213）。

【0071】処理回路C3による処理結果が出力されたときに、データカウンタの値が、入力データの全てのブロック数Nより小さい場合は、再び選択信号S1により処理回路C1を選択することから始めて、すべての入力データを処理し終えるまで、ステップS205からステップS213までの処理サイクルを繰り返す（ステップS214）。

【0072】このように、選択信号による3回の回路情報の選択と、それに引き続くデータ処理により、1ブロックのデータが処理される。このサイクルを入力データのブロック数Nだけ繰り返すことにより、すべての入力データが処理される。

【0073】なお、図34と図35には示されていないが、回路情報の読み込み時や処理回路での処理時にエラーが発生した場合は、アプリケーション制御部にエラーの発生を通知して処理は終了する。

【0074】この例の説明における処理回路C1、C2、C3が中間データを保持する回路構成は、図27、図28、図29、および図30、図31、図32を用いて説明したものと同じ回路構成を利用することができる。

【0075】上述したように、従来のプログラマブル論理回路装置を用いた再構成方法の場合には、一つ一つのブロックごとに、複数個の回路を、順次に、繰り返し再構成する必要があり、再構成の回数が（ブロック数×回路数）分、必要であり、そのため、総処理時間が長くなってしまうという問題がある。

【0076】この発明は、マルチコンテキスト技術のように、回路性能の低下、消費電力の増加、製造コストの増加などをもたらす複数の回路情報を格納するコンフィギュレーションメモリを用いることなく、しかも、ブロック単位の複数の処理を繰り返し行う場合においても、

総合の処理時間を短縮することができるキャッシュロジック技術を実現する方法を提供することを目的とする。

#### 【0077】

【課題を解決するための手段】上記課題を解決するため、この発明によるプログラマブル論理回路装置によるデータ処理方法は、回路素子と、この回路素子に接続されるコンフィギュレーションメモリとを備え、コンフィギュレーションメモリに書き込まれる回路情報に基づいて前記回路素子が用いられて回路が構成されるプログラマブル論理回路部を備えると共に、前記回路情報に基づいて構成された回路で処理されるデータを一時的に保持するデータキャッシュ手段を備えるプログラマブル論理回路装置を用いて、ブロック化されている入力データを、複数個の回路でそれぞれ前記ブロック単位で処理を順次に行うデータ処理方法において、前記複数個の回路のうちの最初の回路の回路情報を前記コンフィギュレーションメモリに書き込み、前記プログラマブル論理回路部に前記最初の回路を構成する第1コンフィギュレーション工程と、前記データキャッシュ手段で保存可能な前記入力データの複数ブロックを、前記最初の回路で処理し、その処理結果のブロック単位のデータを前記データキャッシュ手段に保存する第1処理工程と、前記データキャッシュ手段に、前記入力データの複数ブロック分の処理結果が保存された後に、前記複数個の回路のうちの次の回路の回路情報を前記コンフィギュレーションメモリに書き込み、前記プログラマブル論理回路部に前記次の回路を構成する次コンフィギュレーション工程と、前記データキャッシュ手段に保存された複数ブロック分の処理結果のデータを、前記次コンフィギュレーション工程で構成された回路で処理し、その処理結果の複数ブロック分のデータを前記データキャッシュ手段に保存する次処理工程と、前記次コンフィギュレーション工程と、前記次処理工程とを前記複数個の回路の最後の回路まで繰り返し、前記最後の回路の処理結果を出力データとして外部に出力する出力処理工程と、を備え、前記第1コンフィギュレーション工程から出力処理工程までを、前記入力データの複数ブロック分ごとに、前記入力データの全てについて行うことを特徴とする。

#### 【0078】

【作用】上記のように構成した、この発明のプログラマブル論理回路装置によるデータ処理方法においては、入力データの1ブロック毎に複数個の回路を順次にプログラマブル論理回路装置に再構成して処理を行うのではなく、プログラマブル論理回路装置に設けたデータキャッシュ手段に保存可能な複数個のブロック毎に、複数個の回路を順次にプログラマブル論理回路装置に再構成して処理を行う。

【0079】すなわち、プログラマブル論理回路装置に再構成する回路では、それぞれデータはブロック単位で処理を行うのであるが、入力データの1ブロックの処理

が終了したら次の回路を再構成するのではなく、処理結果の1ブロックのデータをデータキャッシュ手段に格納し、次の1ブロックのデータの処理と同じ回路で行い、その処理結果を続いてデータキャッシュ手段に格納していくというように、複数ブロックの処理を同じ回路で行って、データキャッシュ手段にその処理結果の中間データを保存する。

【0080】そして、最初の回路で、データキャッシュ手段に保存可能なブロック数の処理が終了したら、2番目の回路をプログラマブル論理回路装置に再構成する。この2番目の回路の入力データは、データキャッシュ手段に保存された中間データである。

【0081】そして、2番目の回路においても、データ処理はブロック単位に行うのであるが、データキャッシュ手段に保存された中間データの全てについて処理を行い、そして、この2番目の回路の処理結果は、この2番目の回路が最後の処理回路でなければ、順次にデータキャッシュ手段に中間データとして格納する。このとき、データキャッシュ手段に前に格納されていた中間データは上書きされる。

【0082】そして、2番目の回路によるデータキャッシュ手段に保存されていた複数ブロック分の中間データについての処理が終了すると、第3番目の回路が次にプログラマブル論理回路装置に再構成され、2番目の回路のときと同様にして、そのデータキャッシュ手段に保存されているデータを入力データとして処理をブロック単位で行い、その処理結果をデータキャッシュ手段に格納する。そして、この処理を複数の回路の最後の回路まで繰り返す。

【0083】そして、プログラマブル論理回路部に構成された回路が最後の処理回路であれば、その処理結果は、データキャッシュ手段に保存するのではなく、出力データとしてプログラマブル論理回路装置の外部に出力する。

【0084】入力データの全てのデータについて、以上の複数ブロック単位のデータ処理を行う。

【0085】したがって、この発明によるデータ処理方法の場合には、1ブロック毎に複数個の回路を再構成するのではなく、データキャッシュ手段に保存可能な複数ブロック毎に複数個の回路を再構成するので、前述した従来のプログラマブル論理回路装置によるデータ処理方法に比べて、プログラマブル論理回路装置への回路の再構成回数が少なくなり、その分だけ、データ処理総時間が短縮される。処理単位のブロック数をK個とすれば、プログラマブル論理回路装置への回路の再構成回数は、前述した従来の場合の1/Kになる。

#### 【0086】

【発明の実施の形態】以下、この発明によるプログラマブル論理回路装置によるデータ処理方法およびそれを用いた情報処理システム並びにそれに用いるプログラマ

ル論理回路装置の実施の形態について、図を参照しながら説明する。

【0087】[この発明によるプログラマブル論理回路装置の構造の説明] まず、この発明によるプログラマブル論理回路装置の実施の形態について説明する。

【0088】[第1の構造] この発明によるプログラマブル論理回路装置の第1の構造（プログラマブル論理回路装置の第1の実施の形態）を図1に示す。

【0089】この第1の構造のプログラマブル論理回路装置50は、回路情報を外部から読み込む回路情報入力制御部51と、読み込んだ回路情報に従って回路機能を実現するプログラマブル論理回路部52と、処理途中の中間データを一時的に保持するデータキャッシュ部53とから構成される。

【0090】プログラマブル論理回路部52は、回路素子52Cとコンフィギュレーションメモリ52Mとを備え、回路情報入力制御部51からコンフィギュレーションメモリ52Mに書き込まれた回路情報を従って回路素子52Cの機能と結線状態がきめられる。

【0091】この第1の構造において、コンフィギュレーションメモリ52Mに読み込まれた回路情報により構成される回路は、目的とする処理回路521に加えて、処理回路521への入力を、外部からの入力データとするか、データキャッシュ部53に保持されたデータとするかで切り替える入力データ切替回路522と、処理回路521からの出力を、外部への出力データとするか、データキャッシュ部53に保持されるデータとするかで切り替える出力データ切替回路523とを備える。

【0092】このプログラマブル論理回路装置50を用いて、ブロック化されている入力データを、複数個の回路でそれぞれ前記ブロック単位で処理を順次に行う場合に、前記複数個の回路の最初の回路の回路情報で、入力データ切替回路522と出力データ切替回路523とが生成される。

【0093】そして、複数の回路のうちの処理順序が最初の回路でのデータ処理では、入力データ切替回路522は、外部からの入力データを処理回路521に供給し、出力データ切替回路523は、処理回路521の出力データをデータキャッシュ部53に供給するように切り替えられる。

【0094】また、複数の回路のうちの処理順序が中間の回路でのデータ処理では、入力データ切替回路522は、データキャッシュ部53からのデータを処理回路521に入力し、出力データ切替回路523は、処理回路521の出力データをデータキャッシュ部53に供給するように切り替えられる。

【0095】また、複数の回路のうちの処理順序が最後の回路でのデータ処理では、入力データ切替回路522は、データキャッシュ部53からのデータを処理回路521に入力し、出力データ切替回路523は、処理回路

521の出力データを外部に出力するように切り替えられる。

【0096】これらの入力データ切替回路522および出力データ切替回路523の切り替え制御は、後述するように、データ処理の流れを管理制御するアプリケーション制御部により行われる。

【0097】[第2の構造] この発明によるプログラマブル論理回路装置の第2の構造（プログラマブル論理回路装置の第2の実施の形態）を、図2に示す。

【0098】この第2の構造のプログラマブル論理回路装置60は、回路情報を外部から読み込む回路情報入力制御部51と、読み込んだ回路情報に従って回路機能を実現するプログラマブル論理回路部52と、処理途中の中間データを一時的に保持するデータキャッシュ部53と、処理回路への入力データを、外部からの入力データとするか、データキャッシュ部53に保持されたデータとするかで切り替える入力データ切替部61と、処理回路からの出力を、外部への出力データとするか、データキャッシュ部53に保持されるデータとするかで切り替える出力データ切替部62とから構成される。

【0099】この第2の構造のプログラマブル論理回路装置60と前述の第1の構造のプログラマブル論理回路装置50との違いは、第1の構造のプログラマブル論理回路装置50では、回路情報によって構成された入力データ切替回路522および出力データ切替回路523と同じ機能を持った入力データ切替部61および出力データ切替部62を、第2の構造のプログラマブル論理回路装置60では、プログラマブル論理回路部52とは別体として備えることである。

【0100】そして、この第2の構造の場合には、コンフィギュレーションメモリ52Mに読み込まれる回路情報は、処理回路521についての情報のみとなる。

【0101】この第2の構造のプログラマブル論理回路装置60を用いる場合には、入力データ切替部61と、出力データ切替部62とは、回路情報によって構成されるものではないので、第1の構造のプログラマブル論理回路装置50に比べて、相対的に回路情報量が小さくなり、より一層の再構成時間短縮が可能となるという利点がある。

【0102】[第3の構造] この発明によるプログラマブル論理回路装置の第3の構造（プログラマブル論理回路装置の第2の実施の形態）を図3に示す。

【0103】この第3の構造のプログラマブル論理回路装置70は、回路情報を外部から読み込む回路情報入力制御部51と、読み込んだ回路情報に従って回路機能を実現するプログラマブル論理回路部52とから構成される。

【0104】この第3の構造においては、コンフィギュレーションメモリ52Mに読み込まれた回路情報により構成される回路が、目的とする処理回路521と、データ

タキャッシュ回路524と、処理回路521への入力を、外部からの入力データとするか、データキャッシュ回路524に保持されたデータとするかで切り替える入力データ切替回路522と、処理回路521からの出力を、外部への出力データとするか、データキャッシュ回路524に保持されるデータとするかで切り替える出力データ切替回路523とを備える。

【0105】この第3の構造のプログラマブル論理回路装置70と、第1の構造のプログラマブル論理回路装置50との違いは、第1の構造のデータキャッシュ部53の機能が、回路情報によって構成されたデータキャッシュ回路524により実現されることである。

【0106】この第3の構造のプログラマブル論理回路装置70は、図17に示した従来からある汎用のプログラマブル論理回路装置を用いて構成することができるというメリットがある。

【0107】【情報処理システムおよびデータ処理方法の実施の形態の説明】

【第1の実施の形態】この発明による情報処理システムおよびデータ処理方法の実施の形態では、発明が解決しようとする課題の欄で説明したのと同様に、複数個の回路によって、入力データをブロック単位に順次に処理するデータ処理を、上述した第1の構造のプログラマブル論理回路装置50、第2の構造のプログラマブル論理回路装置60、第3の構造のプログラマブル論理回路装置70のいずれかを用いた情報処理システムにより行う。

【0108】以下に説明する例は、回路情報CD1、CD2、CD3が、回路情報入力制御部51を介してプログラマブル論理回路部52のコンフィギュレーションメモリ52Mに順次読み込まれて構成される処理回路C1、C2、C3で、Nブロック（N組）の入力データDin1、Din2、…、DinNを順次処理して、Nブロック（N組）の出力データDout1、Dout2、…、DoutNを得るようとするデータ処理である。

【0109】このデータ処理の形態は、プログラマブル論理回路装置が情報処理システムのアプリケーション制御部により制御されるスレーブモードと、プログラマブル論理回路装置のアプリケーション制御部が情報処理システムのアプリケーション制御部から一時的に制御を移管されるマスターモードのふたつに分けることができる。以下、それぞれのモードにおけるデータ処理について、説明する。

【0110】【スレーブモード】図4は、スレーブモードの形態における情報処理システムの構成例を示すもので、この例は、第1の構造のプログラマブル論理回路装置50を用いた場合である。

【0111】この情報処理システムは、アプリケーション制御部101と、回路情報記憶部102と、入力データ記憶部103と、出力データ記憶部104と、プログラマブル論理回路装置50とからなる。

【0112】アプリケーション制御部101は、データ処理全体を制御する。そして、アプリケーション制御部101は、処理する入力データのブロックを指示する（ブロック番号を指示する）データカウンタ201と、データキャッシュ部53に保持されたデータのブロックを指示する（ブロック番号を指示する）キャッシュカウンタ202とを備える。

【0113】回路情報記憶部102は、この例では、3個の回路情報CD1、CD2、CD3を保持しており、アプリケーション制御部101の指示に従い、プログラマブル論理回路装置50に、いずれかの回路情報を提供する。

【0114】入力データ記憶部103は、入力データDin1、Din2、…、DinNを保持し、アプリケーション制御部101の指示に従って、プログラマブル論理回路装置50に、必要な入力データを入力する。

【0115】出力データ記憶部104は、アプリケーション制御部101の指示に応じて、プログラマブル論理回路装置50から出力される出力データDout1、Dout2、…、DoutNを蓄積する。

【0116】次に、この情報処理システムを用いた上述のデータ処理について、図5のタイミングチャートと、図6～図8のフローチャートを用いて説明する。

【0117】処理が開始されると、アプリケーション制御部101により、入力データのブロックを指示すデータカウンタ201の値が「1」にリセットされ（ステップS301）、プログラマブル論理回路装置50のデータキャッシュ部53に保持されたデータのブロックを指示すキャッシュカウンタ202の値がデータカウンタ201の値にセットされる（ステップS302）。

【0118】次に、アプリケーション制御部101からの指示により、回路情報CD1が回路情報記憶部102から、プログラマブル論理回路装置50の回路情報入力制御部51を介してコンフィギュレーションメモリ52Mに読み込まれて、プログラマブル論理回路部52に処理回路C1が構成される（ステップS303）。これが、図5のタイミングチャートにおいて、最初の「読」で示した読み込み実行状態である。

【0119】処理回路C1がプログラマブル論理回路装置50に構成されると、アプリケーション制御部101により、入力データ切替回路522の入力データ取得先が外部の入力データ記憶部103に設定されて、キャッシュカウンタ202が指示す入力データのブロックDin1が、アプリケーション制御部101からの指示により、入力データ記憶部103から処理回路C1に入力される（ステップS304）。

【0120】データが入力されるとアプリケーション制御部101により、キャッシュカウンタ202の値に「1」が加えられる（ステップS305）。そして、その入力データDin1が処理回路C1で処理され（ステ

ップS 3 0 6）、アプリケーション制御部1 0 1により、出力データ切替回路5 2 3の出力先がデータキャッシュ部5 3に設定されて、処理結果のデータがデータキャッシュ部5 3に出力される（ステップS 3 0 7）。

【0 1 2 1】次に、キャッシュカウンタ2 0 2の値と、後に説明する値kとがアプリケーション制御部1 0 1により比較され（ステップS 3 0 8）、キャッシュカウンタ2 0 2の値が、kよりも小さい場合は、ステップS 3 0 4に戻り、再び、キャッシュカウンタ2 0 2が指示示す入力データのブロック（ここでは、キャッシュカウンタ2 0 2の値が2になっているので、ブロックD i n 2）が、アプリケーション制御部1 0 1からの指示により、入力データ記憶部1 0 3から処理回路C 1に入力され、ステップS 3 0 5からステップS 3 0 7の処理が行われ、その処理結果がデータキャッシュ部5 3に出力される。

【0 1 2 2】そして、ステップS 3 0 4からステップS 3 0 7の動作を、ステップS 3 0 8で、キャッシュカウンタ2 0 2の値が、値kよりも大きくなったと判断されるまで繰り返す。以上の、データの入力から処理回路C 1による処理までが、図5のタイミングチャートにおいて、最初の「データ処理」と示したデータ処理実行状態である。

【0 1 2 3】kの値は、データキャッシュ部5 3のメモリ容量に応じて設定される値で、データキャッシュ部5 3に保持することができる入力データのブロック数を超えない整数値である。一般的には、この値が大きいほど、回路情報の読み込み回数が少なくなり、従来のリコンフィギュラブルコンピューティングの回路再構成に比べて、再構成時間が短くなる。

【0 1 2 4】しかしながら、後に説明するように、複数の処理を並列して行う場合は、それぞれの処理における回路情報の読み込み周期を考慮してkの値を決める必要がある。

【0 1 2 5】キャッシュカウンタ2 0 2の値がkよりも大きくなると（ステップS 3 0 8）、処理回路C 1で処理され、データキャッシュ部5 3に保持されたk個のブロックを、次の処理回路C 2で処理するため、アプリケーション制御部1 0 1により、キャッシュカウンタ2 0 2の値がデータカウンタ2 0 1の値（このときのデータカウンタ2 0 1の値は「1」）にセットされる（ステップS 3 0 9）。

【0 1 2 6】次に、アプリケーション制御部1 0 1からの指示により、回路情報CD 2が回路情報記憶部1 0 2から回路情報入力制御部5 1を介してコンフィギュレーションメモリ5 2 Mに読み込まれて、プログラマブル論理回路部5 2に処理回路C 2が構成される（ステップS 3 1 0）。これが図5のタイミングチャートにおいて、2番目の「読」で示した読み込み実行状態である。

【0 1 2 7】処理回路C 2がプログラマブル論理回路装

置5 0に構成されると、アプリケーション制御部1 0 1により、入力データ切替回路5 2 2の入力データ取得先がデータキャッシュ部5 3に設定されて、キャッシュカウンタ2 0 2が指示示すデータキャッシュ部5 3のデータのブロックD c a 1が、アプリケーション制御部1 0 1からの指示により、データキャッシュ部5 3から処理回路C 2に入力される（ステップS 3 1 1）。

【0 1 2 8】データが入力されると、アプリケーション制御部1 0 1により、キャッシュカウンタ2 0 2の値に「1」が加えられる（ステップS 3 1 2）。そして、その入力データD c a 1が処理回路C 2で処理され（ステップS 3 1 3）、アプリケーション制御部1 0 1により、出力データ切替回路5 2 3の出力先がデータキャッシュ部5 3に設定されて、処理結果のデータがデータキャッシュ部5 3に出力される（ステップS 3 1 4）。

【0 1 2 9】次に、キャッシュカウンタ2 0 2の値と値kとが比較され（ステップS 3 1 5）、キャッシュカウンタ2 0 2の値がkよりも小さい場合は、ステップS 3 1 1に戻り、再びキャッシュカウンタ2 0 2が指示示すデータのブロックが、アプリケーション制御部1 0 1からの指示により、データキャッシュ部5 3から処理回路C 2に入力され、ステップS 3 1 2からステップS 3 1 4の処理が行われ、その処理結果がデータキャッシュ部5 3に出力される。

【0 1 3 0】そして、ステップS 3 1 1からステップS 3 1 4の動作を、ステップS 3 1 5で、キャッシュカウンタ2 0 2の値が、kよりも大きくなったと判断されるまで繰り返す。処理回路C 2へのデータの入力から処理回路C 2による処理までが、図5のタイミングチャートにおいて、2番目の「データ処理」で示すデータ処理実行状態である。

【0 1 3 1】キャッシュカウンタ2 0 2の値がkよりも大きくなると（ステップS 3 1 5）、処理回路C 2で処理され、データキャッシュ部5 3に保持されたk個のブロックを、次の処理回路C 3で処理するため、アプリケーション制御部1 0 1によりキャッシュカウンタ2 0 2の値がデータカウンタ2 0 1の値（このときのデータカウンタ2 0 1の値は「1」）にセットされる（ステップS 3 1 6）。次に、アプリケーション制御部1 0 1からの指示により、回路情報CD 3が回路情報記憶部1 0 2から回路情報入力制御部5 1を介してコンフィギュレーションメモリ5 2 Mに読み込まれて、プログラマブル論理回路部5 2に処理回路C 3が構成される（ステップS 3 1 7）。これが図5のタイミングチャートにおいて、3番目の「読」で示した読み込み実行状態である。

【0 1 3 2】処理回路C 3がプログラマブル論理回路装置5 0に構成されると、アプリケーション制御部1 0 1により、入力データ切替回路5 2 2の入力データ取得先がデータキャッシュ部5 3に設定されて、キャッシュカウンタ2 0 2が指示示すデータのブロックD c a 1が、

アプリケーション制御部101からの指示により、データキャッシュ部53から処理回路C3に入力される（ステップS318）。

【0133】データが入力されると、アプリケーション制御部101により、キャッシュカウンタ202の値に「1」が加えられる（ステップS319）。そして、その入力データDca1が処理回路C3で処理され（ステップS320）、アプリケーション制御部101により、出力データ切替回路523の出力先が外部に設定されて、処理回路C3による処理結果のデータが外部の出力データ記憶部104へ出力データとして出力される（ステップS321）。

【0134】次に、キャッシュカウンタ202の値とkが比較され（ステップS322）、キャッシュカウンタ202の値がkよりも小さい場合は、ステップS318に戻り、再びキャッシュカウンタ202が指示するデータのブロックが、アプリケーション制御部101からの指示により、データキャッシュ部53から処理回路C3に入力され、ステップS319からステップS321までの動作を、ステップS322で、キャッシュカウンタ202の値がkよりも大きくなったと判断されるまで繰り返す。

【0135】キャッシュカウンタ202の値がkよりも大きくなると（ステップS322）、アプリケーション制御部101により、データカウンタ201の値にkが加えられる（ステップS323）。

【0136】次に、データカウンタ201の値と、値Nとが比較され（ステップS324）、データカウンタ201の値がNよりも小さい場合は、ステップS302に戻り、再びキャッシュカウンタ202がデータカウンタ201の値にセットされて、これまでに説明した操作を繰り返す。入力データのすべてのブロックを処理してデータカウンタ201の値がNよりも大きくなると、処理は終了する。

【0137】【マスターモード】次に、マスタモードによる処理を説明する。図9は、マスタモードの形態における情報処理システムの構成例を示すもので、この例も、第1の構造のプログラマブル論理回路装置50を用いた場合である。

【0138】この例の情報処理システムにおいては、アプリケーション制御部101は、プログラマブル論理回路装置50内に、回路情報により構成される。そして、システムのアプリケーション制御部105が設けられ、プログラマブル論理回路装置50で実行される処理に関する制御が、このシステムのアプリケーション制御部105から、プログラマブル論理回路装置50のアプリケーション制御部101に一時的に移管されることにより、処理が開始される。

【0139】処理が開始されると、図9のプログラマブル論理回路装置50に形成されたアプリケーション制御部101により入力データのブロックを指示するデータカウンタ201の値が1にリセットされ、データキャッシュ部53に保持されたデータのブロックを指示するキャッシュカウンタ202の値がデータカウンタ201の値にセットされる。

【0140】次に、前述したスレーブモードによる処理と同じ手順により、回路情報CD1、CD2、CD3が、回路情報記憶部102から回路情報入力制御部51を介してプログラマブル論理回路部52のコンフィギュレーションメモリ52Mに順次読み込まれて構成される処理回路C1、C2、C3で、入力データ記憶部103に格納されたNブロックの入力データDin1、Din2、…、DinNを、順次処理して、Nブロックの出力データDout1、Dout2、…、DoutNを出力データ記憶部104へ出力する。

【0141】処理が終了すると、プログラマブル論理回路装置50の制御が、再びシステムのアプリケーション制御部105へ移る。

【0142】スレーブモードとマスターモードのいずれの場合も、3回の回路情報の読み込みと、それに引き続くデータ処理により、kブロックのデータが処理される。このサイクルを、N/k回、繰り返すことにより、すべての入力データが処理される。

【0143】なお、上述の実施の形態は、使用するプログラマブル論理回路装置が第1の構造のプログラマブル論理回路装置50の場合であるが、第2の構造のプログラマブル論理回路装置60、また、第3の構造のプログラマブル論理回路装置70を用いても、ほぼ同様のデータ処理を行うことができる。

【0144】ただし、第2の構造のプログラマブル論理回路装置60を用いた場合には、回路情報を読み込むことにより構成される入力データ切替回路522と出力データ切替回路523と同じ機能を持った入力データ切替部61と出力データ切替部62とを備えているので、相対的に回路情報が小さくなり、より一層の再構成時間短縮が可能となるというメリットがある。

【0145】また、第3の構造のプログラマブル論理回路装置70を用いた場合には、データキャッシュ部53の機能が、回路情報によって構成されたデータキャッシュ回路524により実現されるので、回路規模の増大に伴う消費電力の増加とチップコストの増加を防止できるというメリットがある。

【0146】前述もしたように、従来の方法では、3回の回路情報の読み込みと、それに引き続くデータ処理により、1ブロックのデータのみしか処理されない。その結果、すべての入力データを処理するためには、このサイクルを、入力データのブロック数Nだけ繰り返すことが必要であった。

【0147】これに比べ、この発明の上述の実施の形態のデータ処理方法では、すべての入力データを処理するために必要なサイクル数は、 $N/k$ である。すなわち、この発明を用いれば、回路情報の読み込み回数は $1/k$ になり、回路の再構成時間を大幅に短縮することが可能となる。

【0148】ところで、第1の構造および第2の構造のプログラマブル論理回路50および60では、データキャッシュ部53として、余分なメモリ領域が必要となる。しかしながら、マルチコンテキスト技術で必要となる余分なコンフィギュレーションメモリと異なり、データキャッシュ部53は、プログラマブル論理回路部52とは別の部分に形成することができる、回路素子52Cの負荷容量の増大に起因する回路性能の低下をもたらすことが無い。

【0149】また、上述の実施の形態の場合、マルチコンテキスト技術で必要となる余分なコンフィギュレーションメモリに比べて容量の小さなメモリでデータキャッシュ部を構成しても、回路の再構成時間短縮の効果が大きいので、回路規模の増大に伴う消費電力の増加とチップコストの増加はわずかである。

【0150】【第1の実施の形態のデータ処理の具体例】上述したこの発明によるデータ処理の第1の実施の形態の具体例として、JPEG (Joint Photographic Expert Group) 画像圧縮処理の場合について、次に説明する。

【0151】(JPEG方式による画像圧縮) 図10に、JPEG方式の画像圧縮処理の基本ブロックダイヤグラムを示す。入力画像は、8×8画素ブロック単位で、DCT変換、量子化、エントロピー符号化が行われて、圧縮データが出力される。

【0152】入力画像データは、8×8画素ブロックに分割され、分割されたブロック毎にDCT変換部81で、2次元DCT変換が行われる。変換の結果、64個のDCT係数が得られる。

【0153】高い符号化効率を得るために、得られた64個のDCT係数は、量子化部82において、量子化テーブル83を用いて、DCT係数毎に異なるステップサイズで線形量子化される。

【0154】量子化された64個の係数は、エントロピー符号化部84で、符号化テーブル85を参照しながらエントロピー符号化されて圧縮データとなる。エントロピー符号化の方式には、ハフマン符号化を用いる。

【0155】この例では、JPEG画像圧縮処理を、DCT変換、量子化、エントロピー符号化の3処理に分けて、上述したこの発明の第1の実施の形態のデータ処理方法を適用する例を示す。すなわち、DCT変換、量子化、エントロピー符号化に対応する回路情報が、それぞれ、回路情報CD1、CD2、CD3であり、構成される回路が、それぞれ、回路C1、C2、C3である。

【0156】(具体例の構成) DCT変換、量子化、エントロピー符号化の各処理の回路規模は、それぞれ、約30Kゲート、約30Kゲート、約40Kゲートである。そこで、本具体例では、57906システムゲートに対応したザイリンクス (Xilinx) 社のXCV50相当のデバイスを用いる。XCV50は、動的に部分再構成可能なFPGA型のプログラマブル論理回路装置であり、コンフィギュレーションメモリ容量は、559232ビットである。

【0157】この発明の第1の実施の形態を適用するにあたり、XCV50相当のデバイスには、コンフィギュレーションメモリの容量と同じ559232ビットのデータキャッシュ部が備えられているものとする。

【0158】また、この発明の効果を従来技術と比較するために、DCT変換、量子化、エントロピー符号化の各処理に対応する3つのコンフィギュレーションメモリ (559232ビット×3=1677696ビット) を備えたマルチコンテキスト技術によるXCV50相当のデバイスも想定することにする。

【0159】処理対象の画像は、1024画素×1024画素×8ビットの画像である。処理単位は8×8画素ブロック (8画素×8画素×8ビット=512ビット/ブロック) であるので、入力データのブロック数は、 $(1024\text{画素} \times 1024\text{画素}) / (8\text{画素} \times 8\text{画素}) = 16384$ ブロックである。

【0160】入力データの精度は8ビットであるが、処理の途中ではデータの桁数が増えるので、データキャッシュ部53またはデータキャッシュ回路524は16ビットでデータを保持することにする。このとき、データキャッシュ部53またはデータキャッシュ回路524が保持することができるデータのブロック数kの最大値は、 $559232\text{ビット} / (8\text{画素} \times 8\text{画素} \times 16\text{ビット}) = 546.125$ となる。本具体例では、 $k = 512$ とする。

【0161】(従来のプログラマブル論理回路装置を用いた場合の再構成時間) はじめに、動的に部分再構成可能なXCV50を用いた従来のリコンフィギュラブルコンピューティングにおける回路構成時間を見積もる。

【0162】既に説明したように、3回の回路情報の読み込み、およびそれに引き続くデータ処理により、1ブロックのデータが処理され、このサイクルを入力データのブロック数Nだけ繰り返すことにより、すべての入力データが処理される。よって、3つの回路情報の読み込み時間を、それぞれ、Tcd1、Tcd2、Tcd3とすると、Nブロックのデータを処理するときに必要な回路構成時間T(従来)は、

$$T(\text{従来}) = (T_{cd1} + T_{cd2} + T_{cd3}) \times N$$

【0163】(マルチコンテキスト技術を用いた場合の再構成時間) 次に、3つのコンフィギュレーションメモリ

リを備えたXCV50相当のデバイスを用いたマルチコンテキスト技術によるリコンフィギュラブルコンピューティングにおける回路構成時間を見積もる。

【0164】既に説明したように、回路情報の読み込みは、処理の始めに1度だけであり、選択信号による3回の回路情報の選択と、それに引き続くデータ処理により1ブロックのデータが処理される。このサイクルを入力データのブロック数Nだけ繰り返すことにより、すべての入力データが処理される。よって、3つの回路選択の時間を、それぞれ、Ts1、Ts2、Ts3とすると、Nブロックのデータを処理するときに必要な回路構成時間T（マルチコンテキスト）は、

$$T(\text{マルチコンテキスト}) = (T_{cd1} + T_{cd2} + T_{cd3}) + (Ts1 + Ts2 + Ts3) \times N$$

となる。

【0165】Ts1、Ts2、Ts3はTcd1、Tcd2、Tcd3に比べて十分に小さいので、

$$T(\text{マルチコンテキスト}) \approx (T_{cd1} + T_{cd2} + T_{cd3})$$

と近似する。

$$\begin{aligned} T(\text{マルチコンテキスト}) &= T(\text{従来}) / N \\ &= T(\text{従来}) / 16384 \\ &\approx 6.1 \times 10^{-5} T(\text{従来}) \end{aligned}$$

$$\begin{aligned} T(\text{この発明}) &= T(\text{従来}) / k \\ &= T(\text{従来}) / 512 \\ &\approx 2.0 \times 10^{-3} T(\text{従来}) \end{aligned}$$

となる。

【0169】マルチコンテキスト技術による方式と、この発明による方式は、どちらも従来方式に比べて回路構成時間を大幅に短縮することが可能であることがわかる。この発明による方式は、従来方式に比べて回路構成時間を、1/512に短縮することができる。マルチコンテキスト技術による方式は、さらにこの発明の方式に比べて、回路構成時間を512/16384=1/32に短縮することができる。

【0170】したがって、回路構成時間のみで比べると、マルチコンテキスト技術による方式の方がこの発明よりも効果的であるが、マルチコンテキスト技術は、次に説明するように、回路性能の低下、消費電力の増加、製造コストの増加という問題点がある。

【0171】（回路性能、消費電力、製造コストの比較）回路性能をプログラマブル論理回路部のRC時定数で、消費電力をプログラマブル論理回路装置を形成する半導体チップ全体の消費電力で、製造コストを同じく半導体チップ製造コストで、それぞれ比較する。これらの値を図11に示したデバイスの面積構成比から見積もる。

【0172】従来の方式によるXCV50相当デバイスの面積構成比は、プログラマブル論理回路部のコンフィギュレーションメモリが約50%、プログラマブル論理

【0166】（この発明の第1の構造のプログラマブル論理回路装置50を用いたデータ処理方法による再構成時間）次に、コンフィギュレーションメモリと同じ容量のデータキャッシュ部を備えたXCV50相当のデバイスを用いた、この発明の第1の構造のプログラマブル論理回路装置50を用いた第1の実施の形態のデータ処理方法によるリコンフィギュラブルコンピューティングにおける回路構成時間を見積もる。

【0167】既に説明したように、3回の回路情報の読み込みと、それに引き続くデータ処理により、kブロックのデータが処理される。このサイクルをN/k回繰り返すことにより、すべての入力データが処理される。よって、Nブロックのデータを処理するときに必要な回路構成時間T（この発明）は、

$$T(\text{この発明}) = (T_{cd1} + T_{cd2} + T_{cd3}) \times N / k$$

である。

【0168】（回路構成時間の比較）本具体例では、N=16384、k=512である。よって、T（従来）を基準にして回路構成時間を比べると、

回路部部の回路素子が約40%、回路情報入力制御部などの制御部が約10%である。

【0173】この面積構成比（コンフィギュレーションメモリ：回路素子：制御部）=（0.5:0.4:0.1）を基準にして、マルチコンテキスト技術と、この発明の第1の実施の形態によるXCV50相当デバイスの面積構成比を次に説明する。

【0174】マルチコンテキスト技術によるXCV50相当デバイスは、プログラマブル論理回路部に3つのコンフィギュレーションメモリを備えるので、（コンフィギュレーションメモリ：回路素子：制御部）=（1.5:0.4:0.1）となり、全体のチップ面積比は2.0となる。

【0175】この発明の第1の実施の形態によるXCV50相当デバイスのプログラマブル論理回路部は従来のものと同じであるが、新たにコンフィギュレーションメモリ相当のデータキャッシュ部を備えるので、（コンフィギュレーションメモリ：回路素子：制御部：データキャッシュ）=（0.5:0.4:0.1:0.5）となり、全体のチップ面積比は1.5となる。

【0176】プログラマブル論理回路部の抵抗Rと容量Cは、回路配線長Lに比例する。回路配線長Lは、およそ回路部面積Sの平方根に比例するので、RC時定数は、 $L^2 = S$ （プログラマブル論理回路部）に比例す

る。

【0177】回路の駆動周波数をf、駆動電圧をVとすると、デバイス全体の消費電力は、 $f \propto V^2$ に比例する。容量Cはデバイス全体の面積Sの平方根に比例するので、消費電力はS（全体）の平方根に比例する。

【0178】製造コストは、デバイス全体の面積に比例するので、製造コストはS（全体）に比例する。

【0179】以上をまとめて、従来のデバイスを基準にしたマルチコンテキスト技術による方法と、この発明の第1の実施の形態による方法の回路性能、消費電力、製造コストの比較結果を図12に示す。

【0180】図12に示したように、プログラマブル論理回路部に余分なコンフィギュレーションメモリを備えるマルチコンテキスト技術による方法は、回路性能が大きく低下し、消費電力と製造コストも増大する。

【0181】一方、プログラマブル論理回路部外にデータキャッシュ部を備えるこの発明による方法は、回路性能の低下はなく、消費電力と製造コストの増加も、マルチコンテキスト技術による方法に比べて小さい。

【0182】マルチコンテキスト技術による方法における回路性能の低下は、次のデータ処理方法の第2の実施の形態で示すような多くのコンフィギュレーションメモリを必要とする場合に顕著となる。

【0183】【データ処理方法の第2の具体例】この発明の第1の実施の形態のデータ処理方法を、同時に2つの処理を並列して行う場合に適用した具体例を、第2の例として説明する。

【0184】この第2の例は、図13に示すように、回路情報CD1、CD2、CD3が回路情報入力制御部51を介してプログラマブル論理回路部52のコンフィギュレーションメモリ52Mに順次読み込まれて構成される処理回路C1、C2、C3で、第1の処理（A処理）を行い、同時に並行して、回路情報CD4、CD5が回路情報入力制御部51を介してプログラマブル論理回路部52のコンフィギュレーションメモリ52Mに順次読み込まれて構成される処理回路C4、C5で、第2の処理（B処理）を行うものである。図13では、データキャッシュ部は省略した。

【0185】（従来の方式による並行処理）A処理とB処理とを別々に従来の方式で行った場合のタイミングチャートを、図14に示す。図14に示した網かけ部分のタイミングでは、A処理とB処理の回路情報の読み込みが同時に発生する。しかしながら、プログラマブル論理回路装置の回路情報入力制御部は、同時に複数の回路情報を読み込むことができない。

【0186】よって、従来の方式で、ふたつの処理を並行して行う場合には、処理に優先順位を付けて、回路情報の読み込みが同時に発生しないようにしなければならない。例えば、B処理を優先して従来の方式で並行処理を行った場合のタイミングチャートを図15に示す。

【0187】この図15からわかるように、A処理は、回路C1による処理が終了しても、B処理の回路情報の読み込みが終了するまで、回路情報CD2を読み込むことができず、実行待ち状態（ストール）が発生する。

【0188】このストールの発生は、A処理の実行時間を著しく長くするという問題を引き起こす。

【0189】（この発明の実施の形態による並行処理）この発明を適用した場合のタイミングチャートを図16に示す。この図16に示すように、データキャッシュ部に格納するデータのブロック数kを適当に選ぶことにより、ストールを発生させること無く、A処理とB処理を並行して実行することができる。その結果、並行処理によりシステム全体の処理速度を向上させることができるとなる。

【0190】（マルチコンテキスト技術による並行処理）マルチコンテキスト技術による方式で、A処理とB処理を並行処理する場合、A処理の回路情報CD1、CD2、CD3と、B処理の回路情報CD4、CD5のすべての組み合わせを備えたコンフィギュレーションメモリが必要となるので、6つのコンフィギュレーションメモリが必要である。その結果、プログラマブル論理回路部の回路規模が著しく増大し、RC時定数が3.7倍になるという処理性能の低下をもたらす。

【0191】これに対して、この発明の実施の形態による方式では、既に説明したように回路性能の低下は発生しない。

【0192】なお、上述の具体例では、1ブロックを構成するデータサンプル数は一定の場合であるが、各ブロックの区切りが判るのであれば、必ずしも、1ブロックが一定データサンプル数で構成されている必要はない。例えば、JPEG圧縮されたデータを伸長復号する場合には、伸長復号の対象の各ブロックは可変長ブロックとなる。

【0193】

【発明の効果】以上説明したように、この発明によれば、従来のマルチコンテキスト技術のように複数の回路情報を記憶するための余分なコンフィギュレーションメモリを必要としないで、プログラマブル論理回路装置の再構成時間を短縮し、回路再構成時間を含めた総処理時間で比べても、ソフトウェア処理に比べて処理時間の短いハードウェア処理を実現できる。その結果、キャッシュロジック技術によるリコンフュギュラブルコンピューティングを用いた、高速で小型化された情報処理装置を実現することができる。

【0194】また、従来のマルチコンテキスト技術のようにプログラマブル論理回路部に余分なコンフィギュレーションメモリを必要としないので、余分なコンフィギュレーションメモリを配置することで生じる配線負荷容量の増大に起因する、回路性能の低下、消費電力の増加をもたらすことなく、マルチコンテキスト技術によるブ

プログラマブル論理回路装置を実現することができる。

【0195】また、ストールを発生させること無く複数の処理を並行処理することが可能となり、その結果、キャッシュロジック技術によるリコンフュギュラブルコンピューティングを用いた、高速で小型化された情報処理装置を実現することができる。

【0196】また、この発明によるプログラマブル論理回路装置は、従来のプログラマブル論理回路装置と同じ構造のプログラマブル論理回路部を用いることができる。従来のプログラマブル論理回路装置の設計を用いて容易に実現することができる。

【0197】また、この発明によるプログラマブル論理回路装置のデータキャッシュ部をプログラマブル論理回路装置に構成することにより、新たに回路装置を作製することなく従来のプログラマブル論理回路装置を用いて、この発明のプログラマブル論理回路装置を実現することができる。

#### 【図面の簡単な説明】

【図1】この発明によるプログラマブル論理回路装置の実施の形態の第1の構造を示す図である。

【図2】この発明によるプログラマブル論理回路装置の実施の形態の第2の構造を示す図である。

【図3】この発明によるプログラマブル論理回路装置の実施の形態の第3の構造を示す図である。

【図4】この発明による情報処理システムの実施の形態の構成例を示す図である。

【図5】この発明によるデータ処理方法の実施の形態の動作を説明するために用いるタイミングチャートを示す図である。

【図6】この発明によるデータ処理方法の実施の形態の動作を説明するために用いるフローチャートの一部を示す図である。

【図7】この発明によるデータ処理方法の実施の形態の動作を説明するために用いるフローチャートの一部を示す図である。

【図8】この発明によるデータ処理方法の実施の形態の動作を説明するために用いるフローチャートの一部を示す図である。

【図9】この発明による情報処理システムの実施の形態の他の構成例を示す図である。

【図10】この発明によるデータ処理方法を適用するデータ処理の具体例を説明するための図である。

【図11】この発明によるデータ処理方法の実施の形態と、従来例とを比較するための図である。

【図12】この発明によるデータ処理方法の実施の形態と、従来例とを比較するための図である。

【図13】この発明によるデータ処理方法の第2の実施の形態を説明するための図である。

【図14】従来のプログラマブル論理回路装置を用いた並列データ処理を説明するためのタイミングチャートで

ある。

【図15】従来のプログラマブル論理回路装置を用いた並列データ処理を説明するためのタイミングチャートである。

【図16】この発明によるデータ処理方法の実施の形態を用いた並列データ処理を説明するためのタイミングチャートである。

【図17】一般的なプログラマブル論理回路装置の構造の例を示す図である。

【図18】図17のプログラマブル論理回路装置のプログラマブル論理回路部の詳細な構造を説明するための図である。

【図19】図17のプログラマブル論理回路装置のプログラマブル論理回路部の詳細な構造を説明するための図である。

【図20】リコンフィグラブルコンピューティング技術の例を示す図である。

【図21】マルチコンテキスト技術に用いられるプログラマブル論理回路装置を説明するための図である。

【図22】図21のプログラマブル論理回路装置のプログラマブル論理回路部の詳細な構成を示す図である。

【図23】マルチコンテキスト技術の用いられるプログラマブル論理回路装置の他の例を説明するための図である。

【図24】従来のリコンフィギュラブルコンピューティングを用いたデータ処理例を説明するための図である。

【図25】図24の例の説明に供するタイミングチャートである。

【図26】図24の例の説明に供するフローチャートである。

【図27】図24の例に用いる回路情報により構成される回路の例を示す図である。

【図28】図24の例に用いる回路情報により構成される回路の例を示す図である。

【図29】図24の例に用いる回路情報により構成される回路の例を示す図である。

【図30】図24の例に用いる回路情報により構成される回路の他の例を示す図である。

【図31】図24の例に用いる回路情報により構成される回路の他の例を示す図である。

【図32】図24の例に用いる回路情報により構成される回路の他の例を示す図である。

【図33】マルチコンテキスト技術を用いたデータ処理例を説明するための図である。

【図34】図33の例の説明に供するタイミングチャートである。

【図35】図33の例の説明に供するフローチャートである。

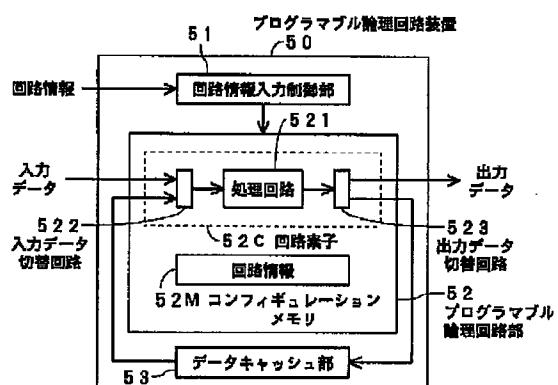
#### 【符号の説明】

50、60、70；プログラマブル論理回路装置

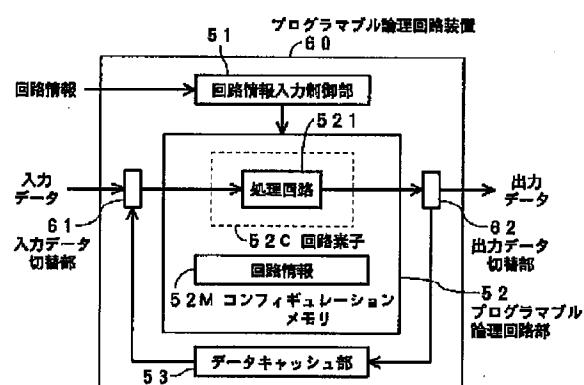
5 1 ; 回路情報入力制御部  
 5 2 ; プログラマブル論理回路部  
 5 2 C ; 回路素子  
 5 2 M ; コンフィギュレーションメモリ  
 5 3 ; データキャッシュ部  
 6 1 ; 入力データ切替部  
 6 2 ; 出力データ切替部  
 1 0 1 ; アプリケーション制御部  
 1 0 2 ; 回路情報記憶部

1 0 3 ; 入力データ記憶部  
 1 0 4 ; 出力データ記憶部  
 1 0 5 ; システムのアプリケーション制御部  
 2 0 1 ; データカウンタ  
 2 0 2 ; キャッシュカウンタ  
 5 2 1 ; 処理回路  
 5 2 2 ; 入力データ切替回路  
 5 2 3 ; 出力データ切替回路  
 5 2 4 ; データキャッシュ回路

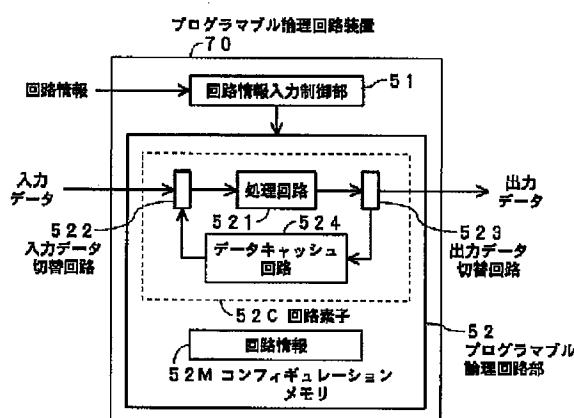
【図1】



【図2】

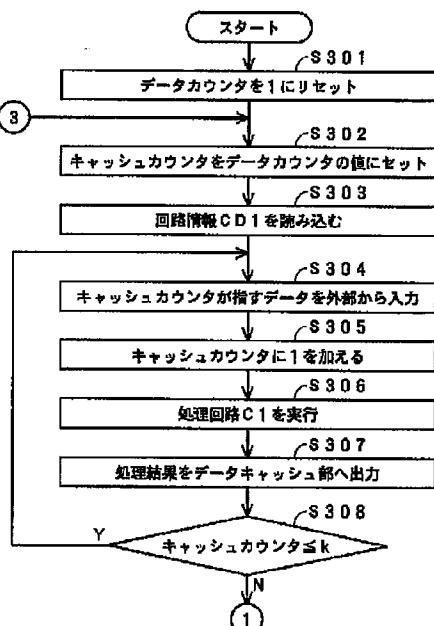


【図3】



【図1.2】

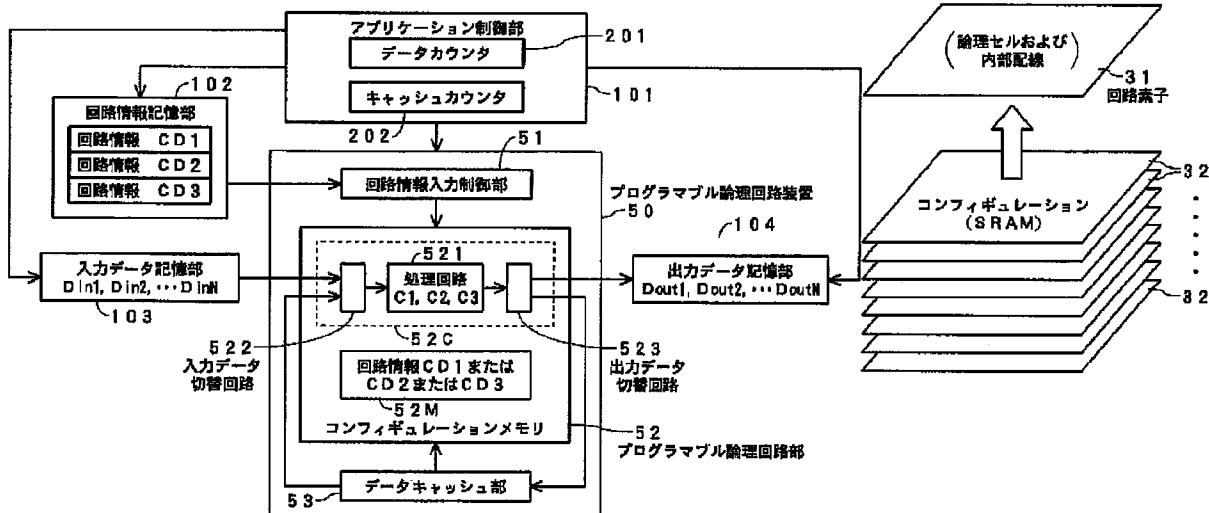
【図6】



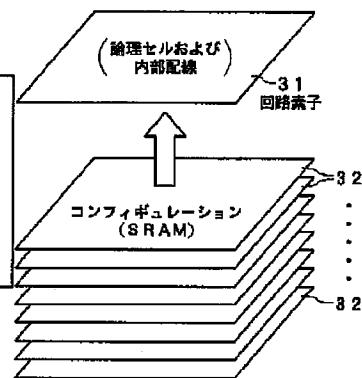
デバイスの回路性能、消費電力、製造コスト

	従来	マルチコンテキスト	本発明
回路性能 (1/R C 時定数)	1	2.1	1
消費電力	1	1.4	1.2
製造コスト	1	2.0	1.5

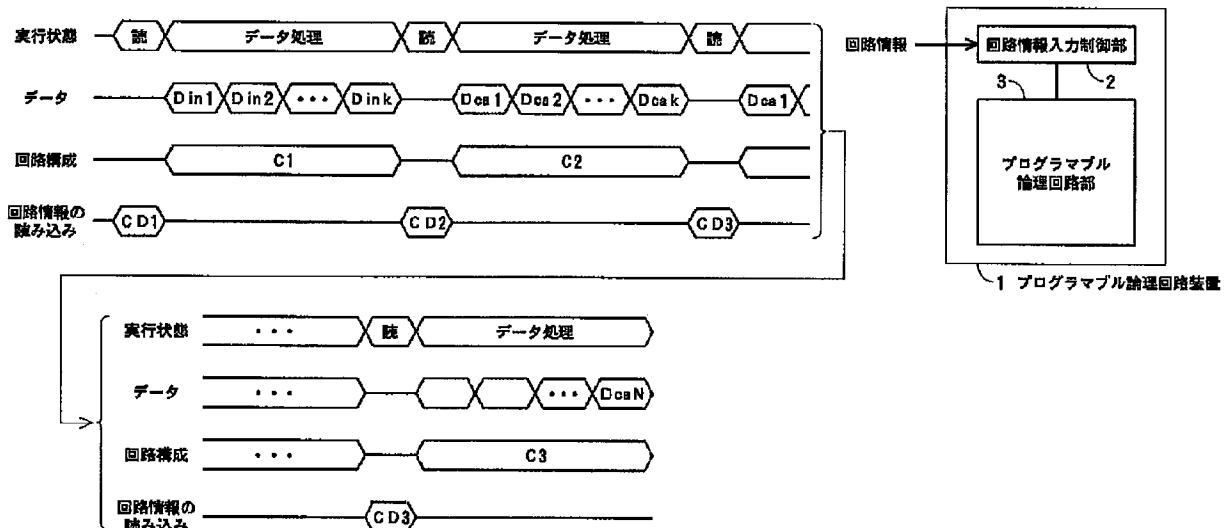
【図4】



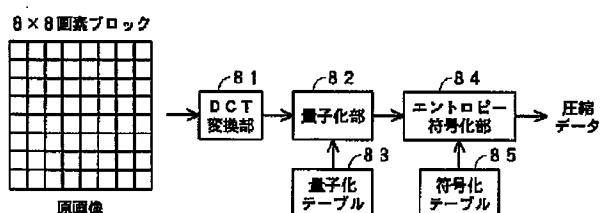
【図23】



【図5】



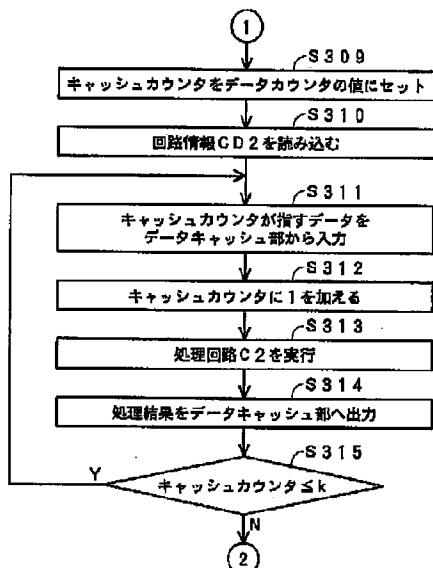
【図10】



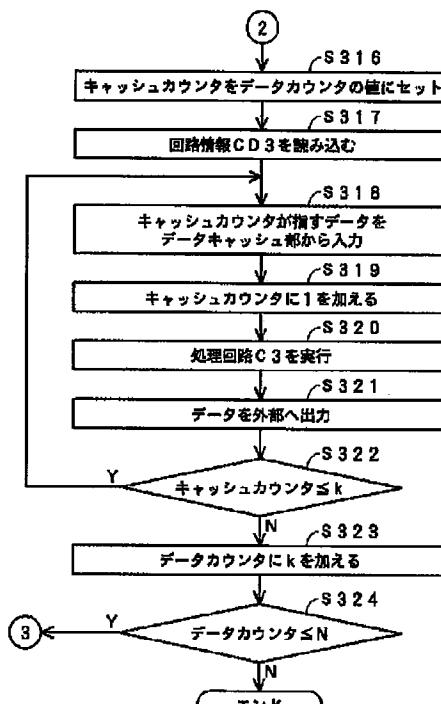
【図11】

デバイスの面積構成比		従来	マルチコンテキスト	本発明
プログラマブル論理回路部	回路素子	0.4	0.4	0.4
コンフィギュレーションメモリ		0.5	1.5	0.5
制御部		0.1	0.1	0.1
データキャッシュ		—	—	0.5
チップ全体		1.0	2.0	1.5

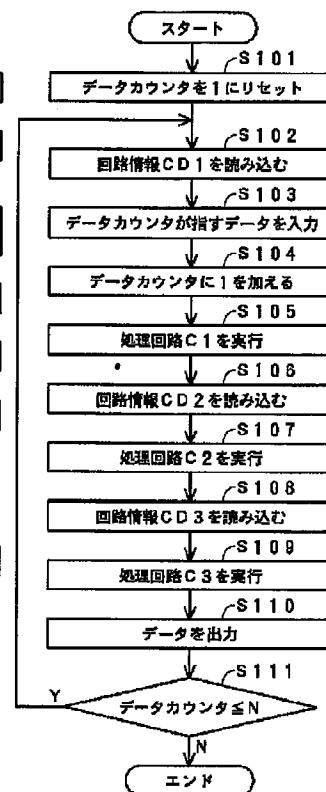
【図7】



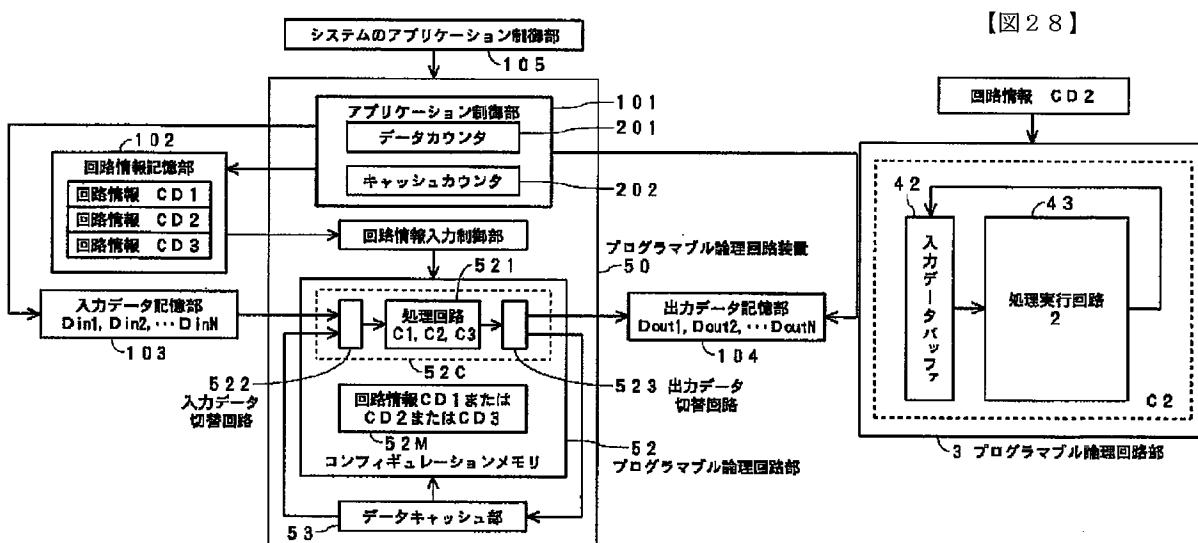
【図8】



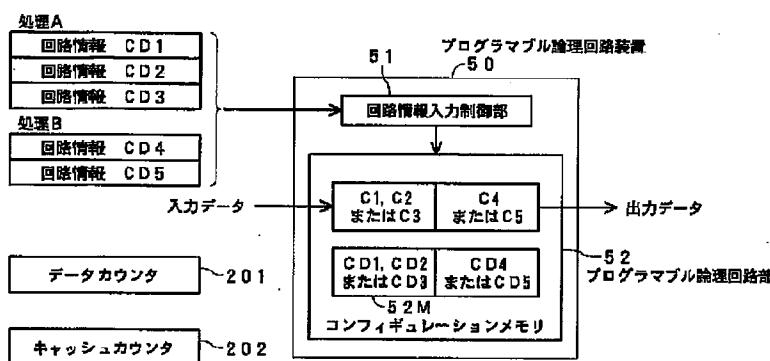
【図26】



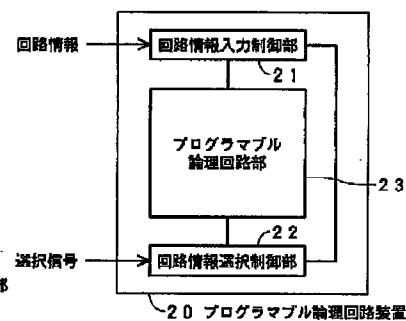
### 【圖 9】



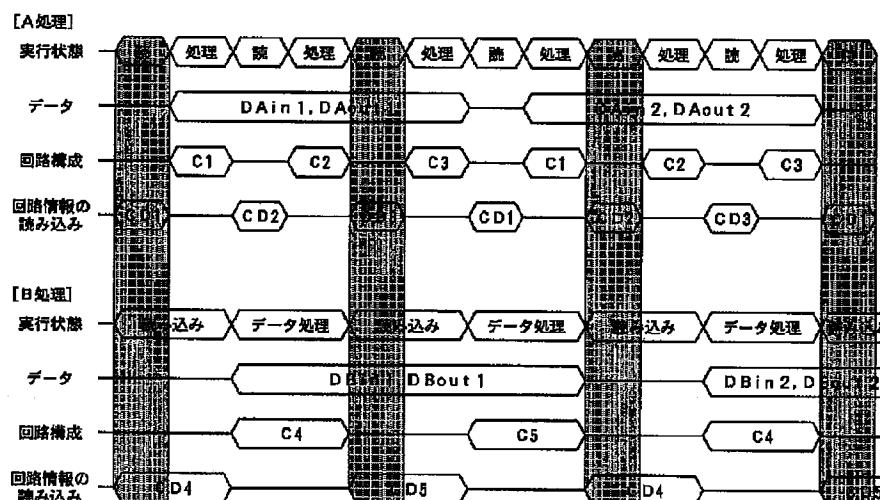
【図13】



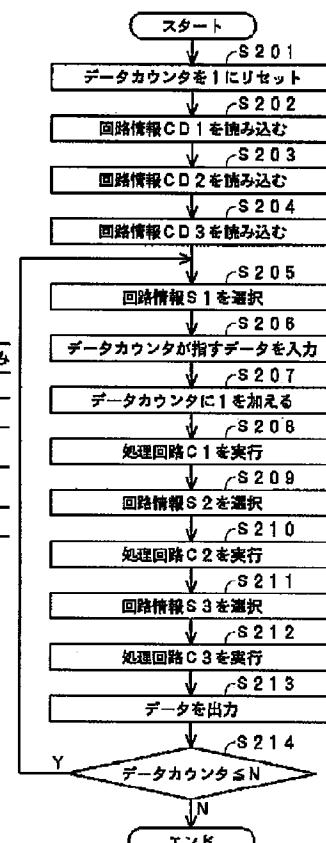
【図21】



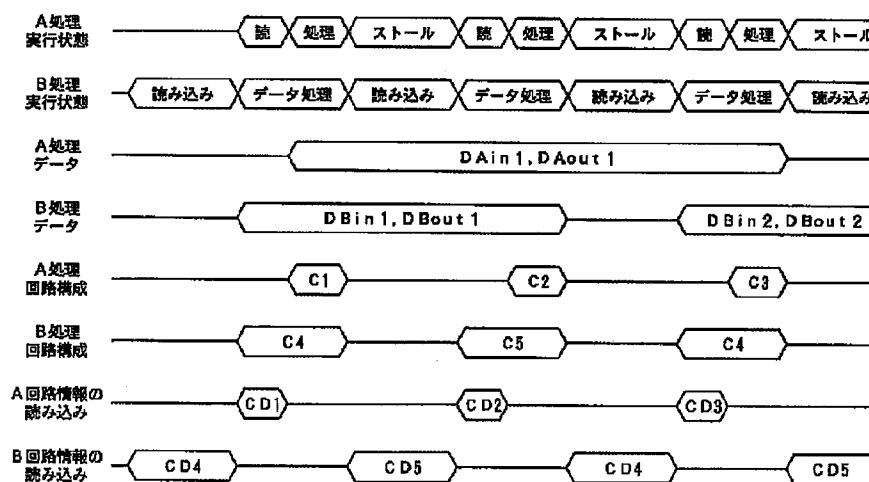
【図14】



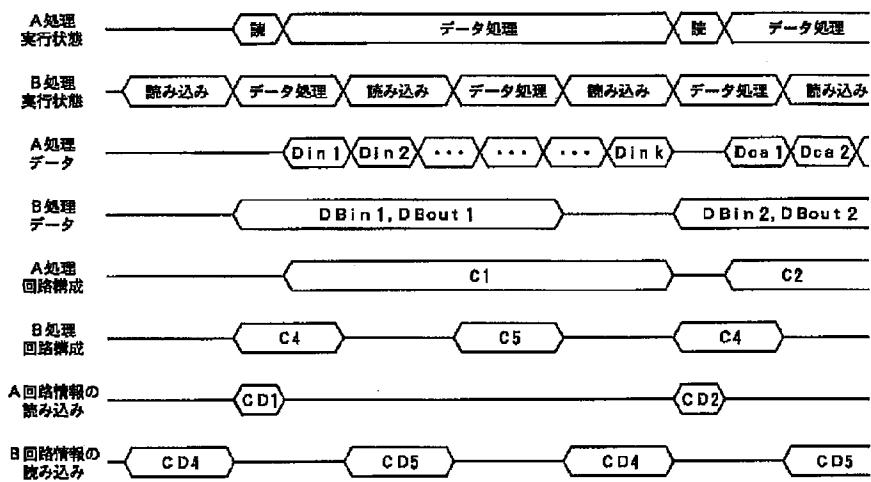
【図35】



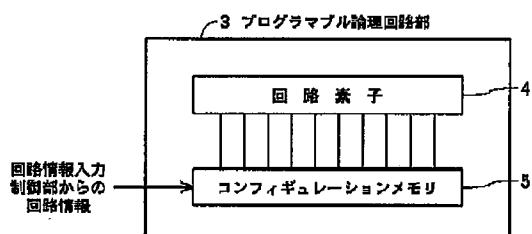
【図15】



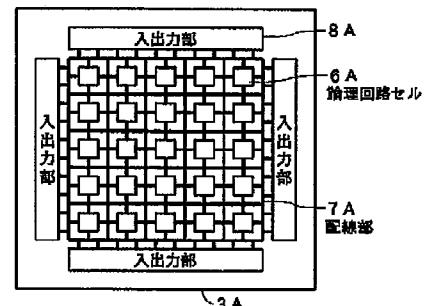
【図16】



【図18】

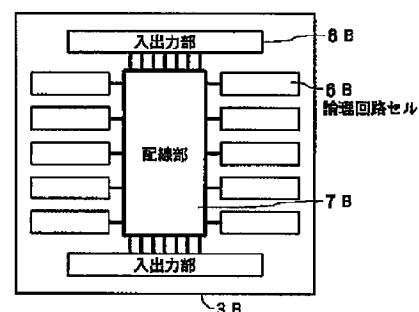


(A)

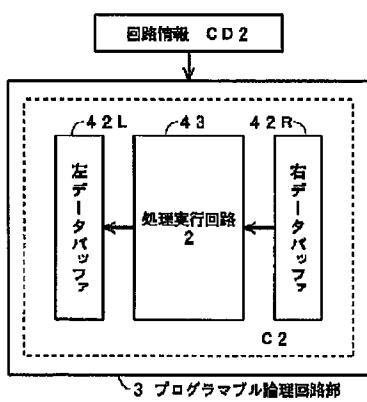


【図19】

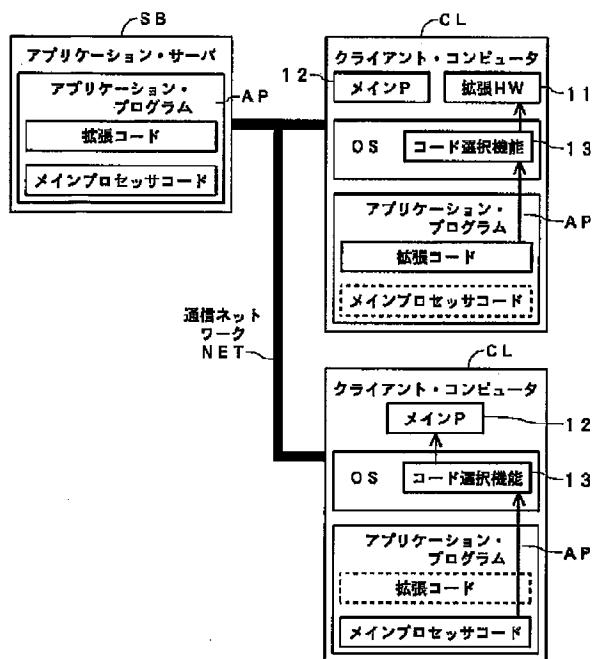
(B)



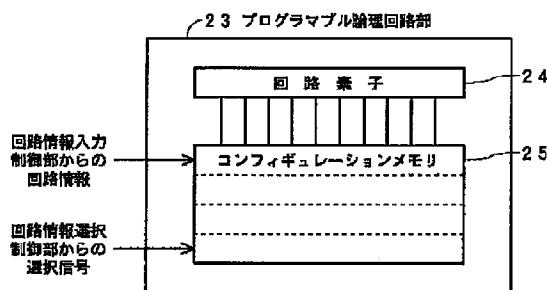
【図31】



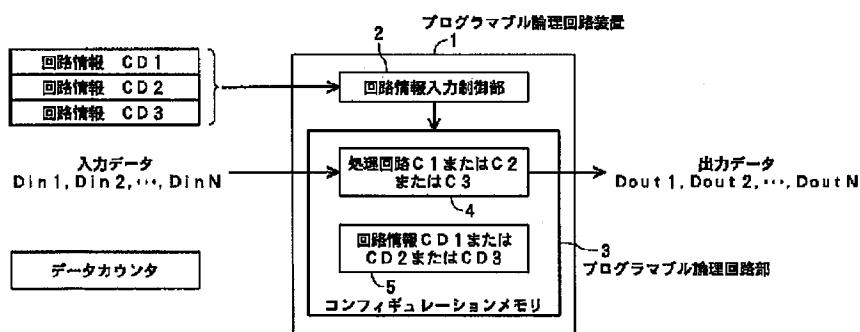
【図20】



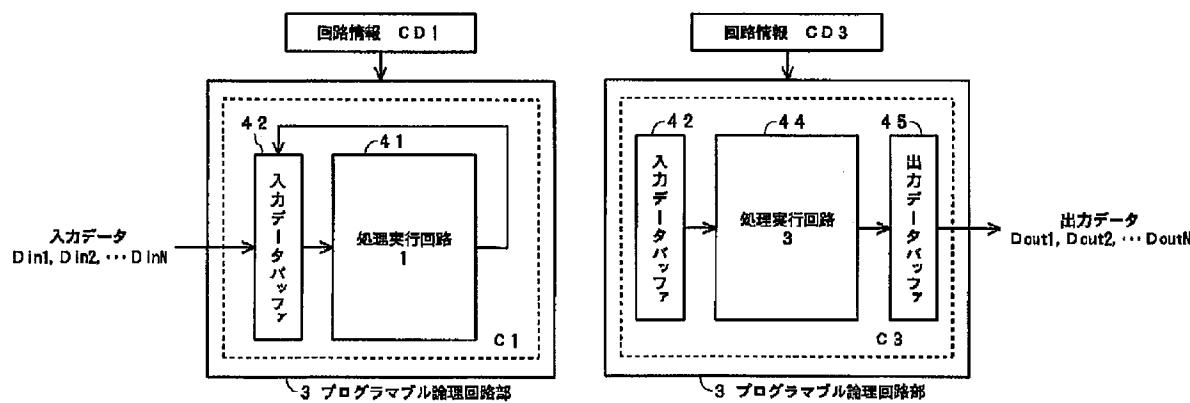
【図22】



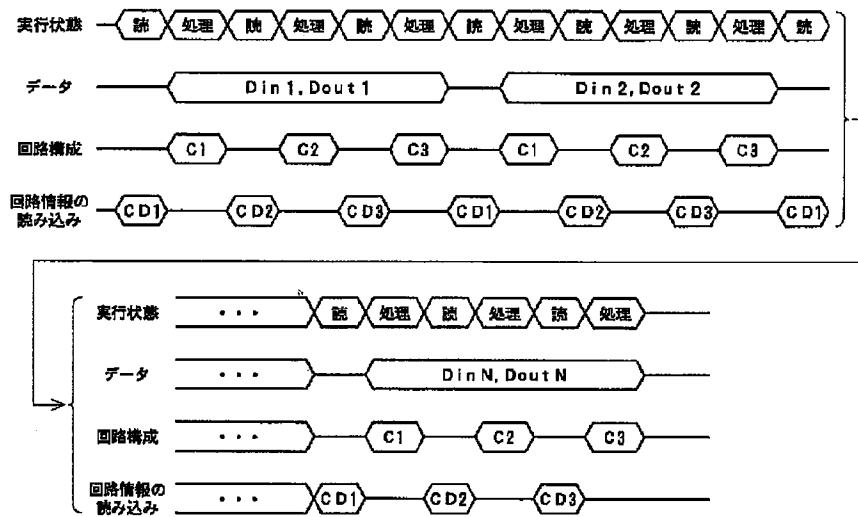
【図24】



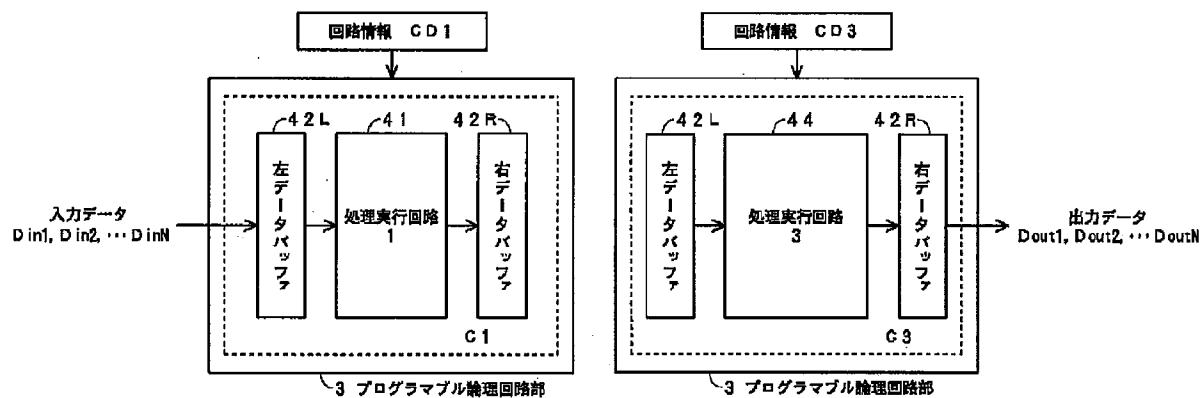
【図27】



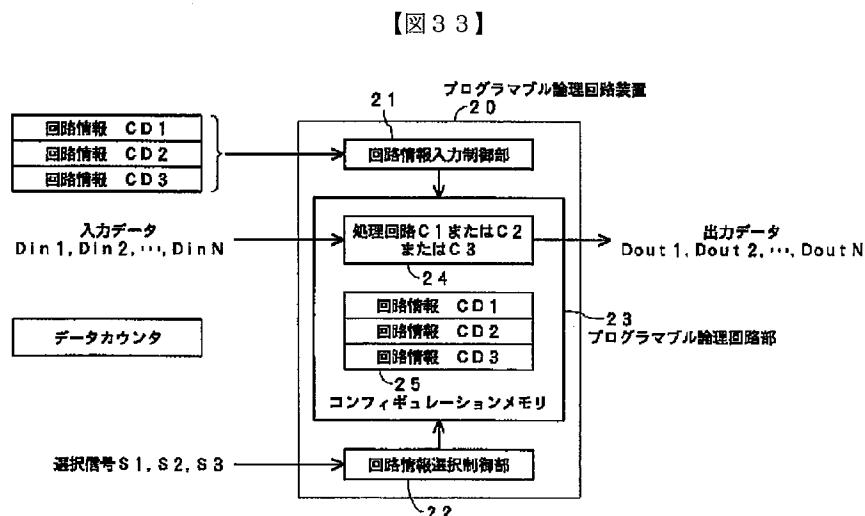
【図25】



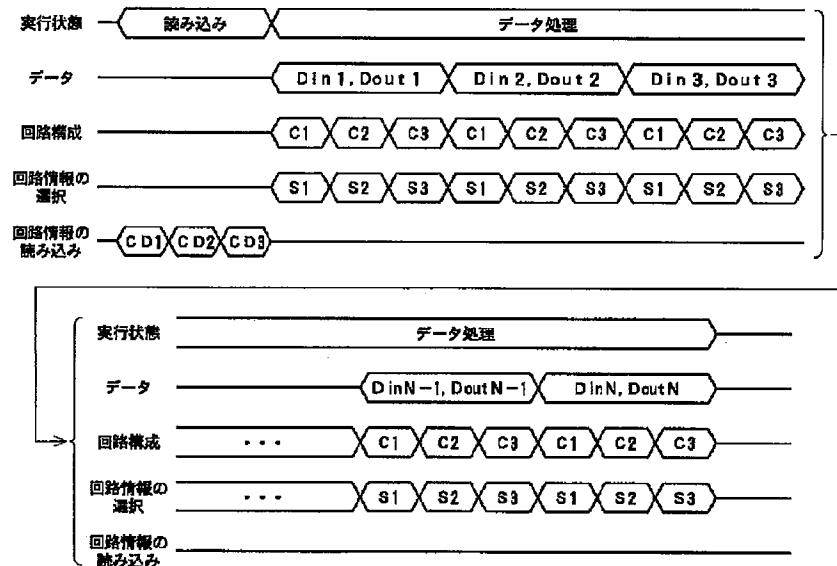
【図30】



【図32】



【図34】



フロントページの続き

(72)発明者 山田 紀一

神奈川県足柄上郡中井町境430 グリーン  
テクなかい富士ゼロックス株式会社内

(72)発明者 里永 哲一

神奈川県足柄上郡中井町境430 グリーン  
テクなかい富士ゼロックス株式会社内

F ターム(参考) 5B076 EB03

5J042 BA04 BA09 CA16 CA20 CA28  
DA00 DA01